UNIVERSITY OF CALIFORNIA SAN DIEGO

Enhanced Meshfree Methods for Numerical Solution of Local and Nonlocal Theories of Solid Mechanics

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Structural Engineering with a Specialization in Computational Science

by

Marco Pasetto

Committee in charge:

Professor Jiun-Shyan Chen, Chair
Professor Randolph Bank
Professor Yuri Bazilevs
Professor Michael Holst
Professor Enrique Luco

2019

The dissertation of Marco Pasetto is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

To my family.

TABLE OF CONTENTS

Signature Page iii

Dedication iv

Table of Contents v

List of Figures viii

List of Tables xii

Acknowledgements xiii

Vita xv

Abstract of the Dissertation xvi

Chapter 1 Introduction 1
1.1 Motivation 2
1.2 Objectives 4
1.3 Outline 5

Chapter 2 Literature Review 7
2.1 The reproducing kernel particle method 8
2.2 The peridynamic theory 10
2.3 Waveform relaxation 12

Chapter 3 The reproducing kernel particle method formulation 15
3.1 Lagrangian reproducing kernel approximation 16
3.2 Semi-Lagrangian reproducing kernel approximation 22
3.3 Appendix 3-A: Pseudo-code for constructing RK shape functions . . 28
3.4 Appendix 3-B: Comparison between RK shape functions and uncorrected kernels 32

Chapter 4 A Lagrangian/semi-Lagrangian coupling approach for meshfree modeling 38
4.1 Lagrangian equation of motion 39
4.2 Blending-based reproducing kernel Lagrangian/semi-Lagrangian coupling 41
4.3 von Neumann stability analysis of the Lagrangian/semi-Lagrangian coupling approach 46
4.4 Numerical results 52
4.4.1 Wave propagation in an elastic bar 53
4.4.2 Taylor bar impact 56

4.4.3 Impact-perforation simulation 60
4.5 Appendix 4-A: von Neumann stability analysis expansions 64

Chapter 5 A reproducing kernel enhanced approach for peridynamic solutions . . . 77
5.1 The Peridynamic Theory 78
5.1.1 Node-based meshfree discretization of peridynamic equations 80
5.2 Reproducing Kernel Enhanced Peridynamics 81
5.3 Numerical Results 88
5.3.1 One-dimensional static peridynamic problem 88
5.3.2 Two-dimensional static peridynamic problem 104
5.3.3 Two-dimensional dynamic crack-branching problem 107
5.4 Peridynamic Modified Nodal Integration Weights 112
5.5 Appendix 5-A: Effect of nodal integration on convergence 118
5.6 Appendix 5-B: Reproducing conditions for integration weights in one-dimensional uniform discretizations 119

Chapter 6 Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics 124
6.1 Review of nonlocal vector calculus 125
6.1.1 Nonlocal divergence and gradient 131
6.1.2 Nonlocal interaction operators 133
6.1.3 Nonlocal integral theorems 134
6.2 Variationally consistent integration for high order exactness in weak form peridynamics 134
6.2.1 Model problem 134
6.2.2 Integration constraints 135
6.2.3 Integration constraints for the nonlocal linear bond-based steady-state peridynamic problem 138
6.3 Node-based variationally consistent integration method: symmetrical weights 140
6.4 Numerical example: one-dimensional static peridynamic problem . 142
6.4.1 Uniform discretization 144
6.4.2 Non-uniform discretization 146

Chapter 7 A Waveform Relaxation Newmark Method for Transient RKPM Modelling of Dynamic Problems 149
7.1 The Waveform Relaxation scheme 150
7.2 Stability and convergence analysis 161
7.2.1 Stability 161
7.2.2 Convergence 166
7.3 Numerical results 169
7.3.1 Free vibration of a two mass-spring system 169
7.3.2 One-dimensional dynamic linear nonlocal problem 175

7.3.3 Two-dimensional plate 177
7.4 Appendix 7-A 186
7.5 Appendix 7-B 188
7.6 Appendix 7-C 192

Chapter 8 Conclusions 194
8.1 Conclusions 195
8.2 Recommendations for future work 197

Bibliography 200

LIST OF FIGURES

Figure 3.1: Meshfree discretization with an RK approximation kernel function. 17

Figure 3.2: Spherical (circular in 2D) and brick (rectangular in 2D) support for Φ_a. . . 19

Figure 3.3: Lagrangian and semi-Lagrangian kernels and associated RK shape functions. The shape functions are built using a linear basis and a cubic B-spline kernel. 24

Figure 3.4: RKPM smoothing cells for SCNI and SNNI (two-dimensional case). 27

Figure 3.5: Kernel functions Φ_I and RK shape functions Ψ_I (1D) for a uniform point distribution. 32

Figure 3.6: Zeroth- and first-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (1D) for a uniform point distribution. Constant and linear bases were used in the RK approximation in (a) and (b), respectively. 33

Figure 3.7: Kernel functions Φ_I and RK shape functions Ψ_I (1D) for a non-uniform point distribution. 33

Figure 3.8: Zeroth- and first-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (1D) for a non-uniform point distribution. Constant and linear bases were used in the RK approximation in (a) and (b), respectively. 34

Figure 3.9: A meshfree discretization of a two-dimensional unit square domain. 35

Figure 3.10: Kernel function Φ_I and RK shape function Ψ_I (2D) near the domain boundary. 35

Figure 3.11: Kernel function Φ_I and RK shape function Ψ_I (2D) in the domain interior. . 36

Figure 3.12: Zeroth-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (2D). 36

Figure 3.13: First-order consistency conditions for the kernel functions Φ_I and RK Ψ_I shape functions (2D). 37

Figure 4.1: Deformation of a material body. 39

Figure 4.2: Linear RK shape functions in the undeformed configuration. 44

Figure 4.3: Linear Lagrangian, linear semi-Lagrangian, and coupled RK shape functions in the current configuration obtained through application of a linear displacement field. 45

Figure 4.4: SNNI smoothing points associated with node I in the reference and current configuration. The smoothing points are marked by black crosses. 49

Figure 4.5: SNNI smoothing points associated with node I for the Lagrangian (indicated as black crosses) and semi-Lagrangian (marked by red asterisks) contributions to the coupled smoothed gradient. 50

Figure 4.6: Lagrangian (red), semi-Lagrangian (blue) and transition (green) zones. . . 54

Figure 4.7: Displacement time history for the bar tip end ($X = 1$). 55

Figure 4.8: Displacement time history for the bar midpoint ($X = 0.5$). 55

Figure 4.9: Initial Taylor bar configurations in the coupling approach. The semi-Lagrangian, Lagrangian and transition zones are represented in blue, red, and green, respectively. 58

Figure 4.10: Deformed Taylor bars. The semi-Lagrangian, Lagrangian and transition zones are represented in blue, red, and green, respectively. 58

Figure 4.11: Semi-Lagrangian (blue), transition (green) and Lagrangian (red) RK zones. . 61
Figure 4.12: Tensile damage at panel the exit face. 62
Figure 4.13: Shear damage at panel the exit face. 62
Figure 4.14: Debris clouds due to concrete perforation at $t = 39 \cdot 10^{-5}$ s. 63

Figure 5.1: Meshfree discretization of a two-dimensional unit square domain with background integration grid. 82
Figure 5.2: Meshfree discretization of a 1D domain with Gauss integration points within the neighborhood of each node. 83
Figure 5.3: Meshfree discretization of a two-dimensional unit square domain with integration points within the neighborhood of each node. The dashed lines represent the limits of the neighborhoods of nodes A and B. 84
Figure 5.4: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. 88
Figure 5.5: h-refinement scheme 91
Figure 5.6: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$, using the node-based meshfree approximation with partial volumes (NB Constant (PV)) and linear RK approximation with $a = 1.001h$. 93
Figure 5.7: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^3$ using node-based meshfree with PV (NB Constant (PV)) and linear and quadratic RK approximations for $\alpha = 0$. Gauss integration scheme 1 is used. 94
Figure 5.8: Convergence of the numerical solution of Eq. (5.36) for $u(x) = e^x$ using linear and quadratic RK approximations. Gauss integration scheme 1 for $\alpha = 0$ is used. 96
Figure 5.9: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$, $\alpha = 0$ in the influence function, and linear RK approximation using Gauss integration scheme 1 and different support sizes a. 97
Figure 5.10: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^3$, $\alpha = 0$ in the influence function using Gauss integration scheme 1 and different support sizes a. 98
Figure 5.11: Convergence of the numerical solution of Eq. (5.36) for $u(x) = e^x$, $\alpha = 0$ in the influence function using Gauss integration scheme 1 and different support sizes a. 99
Figure 5.12: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear approximation for $\alpha = 0$ and different numbers of Gauss integration points per cell (NG_{cell}). 100
Figure 5.13: Error norm of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear RK approximation for $\alpha = 0$ for different number of Gauss points in each integration cells. 102
Figure 5.14: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear approximation for $\alpha = 0$ and non-uniform discretizations arising from perturbation factors $\varepsilon = 0.1, 0.2$. Gauss integration schemes 1 is used. . . . 103
Figure 5.15: 2D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. 104

Figure 5.16: Convergence of the numerical solution of Eq. (5.46) for $\mathbf{u}(\mathbf{x}) = (x_1^2, 0)$, $\alpha = 0$ in Eq. (6.82), using the node-based meshfree solution with FV (NB Constant (FV)) and the linear RK approximation. Gauss integration scheme 1 is used. 106
Figure 5.17: 2D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. The pre-notch is indicated by the dashed line. . 107
Figure 5.18: Fracture damage for the node-based meshfree solution. 110
Figure 5.19: Fracture damage for the RK solution. A consistent density matrix was considered. 110
Figure 5.20: Fracture damage for the RK solution. A lumped density matrix was considered. 110
Figure 5.21: Fracture damage for the node-based meshfree solution. 111
Figure 5.22: Fracture damage for the RK solution. A consistent density matrix was considered. 112
Figure 5.23: Fracture damage for the RK solution. A lumped density matrix was considered. 112
Figure 5.24: Solution for one-dimensional peridynamic problem with manufactured solution using a FV integration approach. 116
Figure 5.25: Solution for one-dimensional peridynamic problem with manufactured solution using nodal integration with modified integration weights with quadratic consistency. 116
Figure 5.26: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using node-based piecewise-constant meshfree solution (with PV) and RK approximations. Integration error strongly affects the convergence behavior. 119

Figure 6.1: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega_g$ 142
Figure 6.2: First-order variational consistency condition obtained with imposition of second-order integration constraint, linear RK shape functions and a uniform point distribution 145
Figure 6.3: Second-order variational consistency condition obtained with imposition of second-order integration constraint, quadratic RK shape functions and a uniform point distribution 145
Figure 6.4: Third-order variational consistency condition obtained with imposition of second-order integration constraint, cubic RK shape functions and a uniform point distribution 146
Figure 6.5: First-order variational consistency condition obtained with imposition of first-order integration constraint, linear RK shape functions and a non-uniform point distribution. 147
Figure 6.6: Second-order variational consistency condition obtained with imposition of second-order integration constraint, quadratic RK shape functions and a non-uniform point distribution. 147
Figure 6.7: Third-order variational consistency condition obtained with imposition of third-order integration constraint, cubic RK shape functions and a non-uniform point distribution. 148

Figure 7.1: Space decomposition and iterations over the time windows. 152
Figure 7.2: Information flow for the Gauss-Seidel WR scheme 153

Figure 7.3: Information flow for the Jacobi WR scheme 154
Figure 7.4: Space-time convergence of Gauss-Seidel and Jacobi waveform relaxation method; the dashed line represents the analytical solutions while the continuous lines represent the WR solution at each iteration ν. 161
Figure 7.5: Undamped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement/velocity/acceleration, $\Delta t = 0.01[s]$; and (right) corresponding absolute error. 171
Figure 7.6: L_∞ norm of the error for different values of Δt. The continuous lines represent the error in the displacement solution while the dashed and dotted ones represent the error in the velocity and acceleration, respectively. 172
Figure 7.7: Damped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement; and (right) corresponding absolute error. 174
Figure 7.8: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. 175
Figure 7.9: Nonlocal 1D dynamic problem spatially discretized with RKPM: comparison between Newmark and Jacobi WRN_β method. (Left) Comparison of displacement; and (right) corresponding absolute error. 177
Figure 7.10: Two dimensional plate discretized with discrete points 178
Figure 7.11: RKPM meshfree solution; (a) Horizontal displacement of the tip central point over time for the WRN_β methods, and (b) the error compared with a refined Newmark method 179
Figure 7.12: CPU time vs number of nodes. Newmark and WRN_β methods. Notice that as the system increases, WRN_β methods converge faster than the Newmark method. 180
Figure 7.13: Two dimensional plate discretized with finite elements. The nodes on the edge at $x = 0$ are fixed. The degrees of freedom associated with 2x2 and 8x8 partitioning block sizes are shown. 181
Figure 7.14: Finite Element solution; (a) Horizontal displacement of the tip central point over time for the WRN_β methods, and (b) the error compared with a refined Newmark method. 182
Figure 7.15: Eigenvalues of the error propagation matrix **R** for the 2D plate problem. . . 183
Figure 7.16: CPU time vs number of elements. Newmark and point-wise WRN_β methods. Notice that as the system increases, WRN_β methods converge faster than the Newmark method. 185

LIST OF TABLES

Table 4.1: Run-times for the one-dimensional wave propagation problem. 56

Table 4.2: Dimensions of the deformed Taylor Bar. 59

Table 4.3: Run-times for the Taylor bar impact problem. 59

Table 4.4: Spherical projectile properties. 60

Table 4.5: Projectile exit velocity. 63

Table 4.6: Run-times for the impact-perforation problem. 63

Table 7.1: Spectral radii of the error propagation matrix **R** and number of iterations for for Jacobi WRN$_\beta$ and Gauss-Seidel WRN$_\beta$ and maximum natural frequency for the different methods. $\Delta t = 0.01[s]$. 173

Table 7.2: CPU time for different time windows dimensions. 174

Table 7.3: Maximum natural frequency for different methods. 182

Table 7.4: spectral radii of **R** and number of iterations for increasing number of elements. 184

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Professor J.S. Chen for his continuous support and guidance, and for taking the time to get to know me, both as a researcher and as a person. I have grown a lot by being part of his research group. Thanks are also extended to Professor Randolph Bank, Professor Yuri Bazilevs, Professor Michael Holst, and Professor Enrique Luco for serving on my committee and for sharing their knowledge with me. I would also like to express my gratitude to Dr. David Littlewood and Dr. Pablo Seleson for mentoring me during my research internships.

It is often said that doing a Ph.D. is a lonely journey, so I would also like to thank the people who made it less so, i.e., the friends (and colleagues) who shared this experience with me, both in Professor Chen's group and outside of it. Thanks to Dr. Jacob Koester, Dr. Frank Beckwith, Dr. Qizhi He, Dr. Haoyan Wei, Jonghyuk Baek and Dr. Guohua Zhou for all the research discussions and for all their support. I will fondly remember the time we spent together. Thanks also to Tsung-Hui Huang, Xiaolong He and Karan Taneja for being good colleagues and friends. Thanks to Dr. Edouard Yreux and Dr. Michael Hillman for showing me the ropes when I was just starting in my first year. Thanks to Dr. Simone Sternini, with whom I shared this experience from the very beginning. Thanks also to Dr. Georgios Moutsanidis, Dr. Marco Pigazzini, Ivana Escobar, Francesco Fraternali, Dr. Christian Secchi, Dr. Marco Orecchioni, Martina Zoccheddu, Martina Cicoli, Dr. Martina Belli, Dr. Francesca Mulas, Gloria Faraone, Leticia Nocko, Dr. Andrea Cugno, Dr. Margherita Capriotti, Dr. Lorenzo Ferrari, Dr. Lorenzo Rossini, Dr. Agnese Paci (and many others) for their friendship and for all the memories we created together.

I would also like to sincerely thank my family for standing by me and for believing in me, even when I did not.

The sponsorship of the research presented in Chapters 3 and 4 by the US Army Engineer Research Development Center for UCSD is also greatly acknowledged.

Portions of Chapters 2, 3, 5, and 8 have been published in M. Pasetto, Y. Leng, J.S. Chen, J.T. Foster, P. Seleson, “A reproducing kernel enhanced approach for peridynamic solutions”, *Computer Methods in Applied Mechanics and Engineering*, 340, 2018. The dissertation author was the primary investigator of the included material.

Portions of Chapters 2, 7, and 8 have been published in Marco Pasetto, Haim Waisman, J.S. Chen, “A waveform relaxation Newmark method for structural dynamics problems”, *Computational Mechanics*, 63(6), pp. 1223-1242, 2019. The dissertation author was the primary investigator of this material.

Portions of Chapters 2, 3, 4, and 8 are currently being prepared for submission for publication of the material. M. Pasetto, J. Baek, J.S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, “A Lagrangian/semi-Lagrangian Coupling Approach for Meshfree Modeling of Extremely Large Deformation Problems”. The dissertation author was the primary investigator of this material.

Portions of Chapter 5, 6, and 8 are currently being prepared for submission for publication of the material. M. Pasetto, J.S. Chen, “Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics”. The dissertation author was the primary investigator of this material.

VITA

2019	Ph.D. in Structural Engineering with a Specialization in Computational Science, University of California San Diego, CA
2018	C. Phil. in Structural Engineering, University of California San Diego, CA
2013	M. S. in Civil Engineering and Engineering Mechanics, Columbia University, New York, NY
2013	Laurea Magistrale (M. S.) in Civil Engineering *cum laude*, Alma Mater Studiorum - Università di Bologna, Italy
2011	Laurea (B. S.) in Civil Engineering *cum laude*, Alma Mater Studiorum - Università di Bologna, Italy

PUBLICATIONS

JOURNAL ARTICLES

Michael Hillman, Marco Pasetto, Guohua Zhou, “Generalized Reproducing Kernel Peridynamics: Unification of Local and Non-Local Meshfree Methods, and an Arbitrary-order Accurate Generalized State-Based Peridynamic Formulation”, *Computational Particle Mechanics*, in press, 2019

Marco Pasetto, Haim Waisman, J.S. Chen, “A waveform relaxation Newmark method for structural dynamics problems”, *Computational Mechanics*, 63(6), pp. 1223-1242, 2019.

Marco Pasetto*, Yu Leng*, Jiun-Shyan Chen, John T. Foster, Pablo Seleson, “A reproducing kernel enhanced approach for peridynamic solutions”, *Computer Methods in Applied Mechanics and Engineering*, 340, pp. 1044-1078, 2018, *contributed equally.

TECHNICAL REPORTS

Marco Pasetto and David J. Littlewood, “Semi-Lagrangian RKPM and its Implementation in the Sierra/Solidmechanics Analysis Code”, in Center for Computing Research Summer Proceedings 2016, J.B. Carleton and M.L. Parks, eds., Technical Report SAND2017-1294R, Sandia National Laboratories, pp. 241-253, 2016.

ABSTRACT OF THE DISSERTATION

Enhanced Meshfree Methods for Numerical Solution of Local and Nonlocal Theories of Solid Mechanics

by

Marco Pasetto

Doctor of Philosophy in Structural Engineering with a Specialization in Computational Science

University of California San Diego, 2019

Professor Jiun-Shyan Chen, Chair

Achieving good accuracy while keeping a low computational cost in numerical simulations of problems involving large deformations, material fragmentation and crack propagations still remains a challenge in computational mechanics. For these classes of problems, meshfree discretizations of local and nonlocal approaches, have been shown to be effective as they avoid some of the common issues associated with mesh-based techniques, such as the need for remeshing due to excessive mesh distortion. Nonetheless, other issues remain.

In the framework of local mechanics, the semi-Lagrangian reproducing kernel particle method (RKPM) has been proved to be particularly effectively for material damage and fragmentation, as by reconstructing the field approximations in the current configuration it does not require the deformation gradient to be positive definite. This, however, results in a high computational cost.

Furthermore, for crack propagation problems, the use of classical local mechanics presents many challenges, such as the need of accurately representing the singular stress field at crack tips. The peridynamic nonlocal theory circumvents these issues by reformulating solid mechanics in terms of integral equations. In engineering applications, a simple node-based discretization of peridynamics is typically employed. This approach is limited to first order convergence and often lacks the symmetry of interaction of the continuous form. The latter can be recovered through the use of the peridynamic weak form, which however involves costly double integration.

First, we first propose, in the context of local mechanics, a blending-based spatial coupling scheme to transition from the computationally cheaper Lagrangian RKPM to the semi-Lagrangian RKPM. Next, we introduce an RK approximation to the field variables in strong form peridynamics to increase the order of convergence of peridynamic numerical solutions. Then, we develop an efficient n-th order symmetrical variationally consistent nodal integration scheme for RK enhanced weak form peridynamics.

Lastly, we propose a Waveform Relaxation Newmark algorithm for time integration of the semi-discrete systems arising from meshfree discretizations of local and nonlocal dynamics problems. This scheme retains the unconditional stability of the implicit Newmark scheme with the advantage of the lower computational cost of explicit time integration schemes.

Numerical examples demonstrate the effectiveness of the proposed approaches.

Chapter 1

Introduction

1.1 Motivation

Achieving good accuracy while keeping a low computational cost in numerical simulations of problems involving large deformations, crack propagations or material fragmentation still remains a challenge in computational mechanics. For this class of problems, meshfree methods have been shown to avoid some of the common issues associated with mesh-based techniques (e.g., the finite element method), such as mesh entanglement and the need for re-meshing due to excessive mesh distortion. Nonetheless, other issues remain.

In the framework of classical local mechanics, the semi-Lagrangian framework reproducing kernel particle method has been proved to be particularly suitable for problems involving material damage and fragmentation. This approach circumvents the issues associated with the use of Lagrangian approaches, which break down when the topology of the problem changes so that the mapping between the undeformed reference configuration and deformed current configuration is no longer one-to-one, by reconstructing the field approximations based on the current configuration (i.e., at every time step of the simulation). While doing so allows for approximation completeness of arbitrary order in the current configuration, reconstructing the RK shape functions and shape function gradients associated with the approximation significantly increases the overall computational cost.

Overall, modeling arbitrary crack growth is very challenging. For problems involving crack propagation, meshfree methods allow representing arbitrary crack geometries by relatively simple approaches such as the visibility criterion, without the need of re-meshing to modify the connectivity between the nodes. However, an accurate representation of the singular stress field occurring at a crack tip requires the use of enrichment techniques. The approximation space can be locally enriched either extrinsically or intrinsically. Extrinsic enrichment techniques introduce additional degree of freedoms, which increases the computational cost, while intrinsic enrichment on subregions of the domain requires careful construction to avoid a discontinuous approximation space at the subdomain interfaces. To circumvent the issues associated with the use of local (i.e., PDE-based) mechanics for crack propagation problems, such as the need to represent singular stress fields, the peridynamic nonlocal theory was developed. Peridynamics is an integro-differential nonlocal reformulation of continuum mechanics, which differently from the local approach does not make differentiability assumptions on the displacement field. For this reason, it remains valid in the presence of discontinuities and it is naturally well-suited for modeling fractures and crack growth. For the aforementioned reasons meshfree methods are commonly used to discretize peridynamic problems. The most common discretization method for peridynamic models used in engineering problems is a node-based meshfree approach. This method discretizes peridynamic domains by a set of nodes, each associated with a nodal cell with a characteristic volume, leading to a particle-based description of continuum systems and approximates the peridynamic integral by nodal integration. The behavior of each particle is then considered representative of its cell. While this leads to a method characterized by a relatively simple implementation and relatively low computational cost, compared to other discretizations, its convergence rate is limited to first-order, making it hard to increase accuracy without significant discretization refinement (and related increased cost).

Furthermore, due to its nonlocal nature, peridynamics is based on the interaction between couple of points in the domain, which is typically symmetrical in nature. However, this symmetry is often lacking in discrete strong peridynamic formulations. This issue can be circumvented through the use of weak form peridynamics. However, since the variational form of the peridynamic balance of linear momentum entails the solution of a double integral, it requires higher computational expense relative to the strong form, which has limited its use for practical applications.

Lastly, once a semi-discrete system is obtained for the meshfree discretization of either the local or the nonlocal approach, time integration is performed. One of the most well known and widely used family of direct time integration methods is the Newmark family of methods.

Its implicit implementation is unconditionally stable but requires the solution of a linear system, which makes it computationally expensive; its explicit form, on the other hand, has a low computational cost as it doesn't require the solution of systems but is conditionally stable, thus limiting the allowed time increment. For problems involving large deformations and fragmentation, such as impact problems, the meshfree particles can often get very close to each other during the impact phase, leading to the need of very small time steps to retain stability in the explicit case. For long simulations this can also lead to an increase of computational cost.

1.2 Objectives

The objective of this work is to enhance the computational efficiency and accuracy of meshfree approaches employed in the solution of local and nonlocal problems, in the framework of large deformation, fracture and material fragmentation problems. To this end, the major developments of this dissertation can be summarized as follows:

- Development of a Lagrangian/semi-Lagrangian coupling scheme to spatially blend the Lagrangian and the semi-Lagrangian RK approximations. This allows for the use of the semi-Lagrangian RK approximation only in those portions of the problem domain where required by the nature of the deformation (e.g., where fragmentation occurs) and to avoid the re-computation of the RK shape functions in the rest of the domain (i.e., where the Lagrangian RK approximation is employed). This results in computational saving, which is assessed in large deformation and fragmentation problems, such as a Taylor bar impact problem or as a projectile perforating a concrete panel.

- Development of an RK enhanced approach for peridynamic solutions. The use of the RK approximation in the field variables in the peridynamic equations, when paired with accurate numerical integration, is shown to increase the order of convergence of peridynamic numerical solutions through manufactured one-dimensional and two-dimensional peridynamic problems.

- Identification of variationally consistent *n*-th order integration constraints for RK enhanced weak form peridynamics and development of an *n*-th order variationally consistent nodal integration scheme. Achieving higher-order accurate integration in a node-based framework should aid towards the use of coarser discretization integration and a lower computational cost.

- The introduction and analysis of a Waveform Relaxation Newmark (WRN_{β}) algorithm for the solution of linear second-order hyperbolic systems of ODEs in time, which retains the unconditional stability of the implicit Newmark scheme with the advantage of the lower computational cost of explicit time integration schemes. The performance of this algorithm is compared to a standard implicit Newmark method on a suite of numerical examples.

1.3 Outline

The remainder of this dissertation is organized as follows. Overviews of the reproducing kernel particle method (RKPM), of the nonlocal theory of peridynamics, and of the waveform relaxation method are given in Chapter 2. The construction of the RK approximation functions, both in the Lagrangian and semi-Lagrangian framework are reviewed in Chapter 3. The advantages and disadvantages of these formulations are then discussed in Chapter 4, where a Lagrangian/semi-Lagrangian RK spatial coupling scheme is also introduced. The stability of the coupled approach is analyzed and several numerical examples are presented to show the ability of the proposed coupling scheme to reduce the computational cost of RKPM meshfree large deformation simulations with respect to pure semi-Lagrangian RK simulations. In Chapter 5, we review the nonlocal peridynamic theory in its strong form and the associated typically used node-based discretization scheme. We then introduce a reproducing kernel enhanced strong-form peridynamic approach, with the goal of improving the convergence behavior in the solution of peridynamic problems. Its ability to improve convergence with respect to the common node-based discretization is demonstrated through convergence studies for 1D and 2D bond-based static linear peridynamic problems. Chapter 6 reviews the weak form of peridynamics through the nonlocal vector calculus framework. n-th order integration constraints for a variationally consistent RK enhanced weak form peridynamics are identified and a modified nodal integration approach to achieved n-th order variational consistency is also presented. In Chapter 7 we the general waveform relaxation Gauss-Seidel and Jacobi relaxation schemes are reviewed and we introduce the Newmark Waveform Relaxation algorithm, with the goal of improving the time-integration efficiency of the semi-discrete equations coming from the discretization of either the classical local or the nonlocal theory. Stability analysis and convergence studies are also performed, followed by discussion on the necessary conditions needed for stability and convergence of the method. The performance of the method in terms of accuracy and computational time is illustrated on the semi-discrete equations of classical structural dynamics and of nonlocal approaches, discretized by the reproducing kernel particle method. Lastly, a summary of this work and concluding remarks are given in Chapter 8.

Chapter 2

Literature Review

2.1 The reproducing kernel particle method

Meshfree methods have been shown to be well-suited for problems where mesh-based approaches, such as the finite element method (FEM), perform poorly due to excessive mesh distortion or entanglement issues. This is the case, for example, of problems involving very large deformations [58][57]. Furthermore, they are suited for fracture modeling [124][122][43], fragment-impact problems [98], and adaptive refinement computations, where re-meshing in mesh-based approaches can become expensive.

In this thesis, we focus on the reproducing kernel particle method (RKPM) [138][56], which is a meshfree approach based on a kernel function to control locality and smoothness, and a correction function to enforce polynomial reproduction up to arbitrary order. RKPM has been formulated as a Lagrangian approach [56][57] and as a semi-Lagrangian approach [55] for modelling of nonlinear solids and structures.

In the RKPM Lagrangian meshfree approach, the variational equations can be formulated in the reference undeformed configuration or in the current deformed one (total Lagrangian approach and updated Lagrangian approach, respectively) and the RK approximation functions are constructed in the reference configuration. Because of the meshfree nature of the method, materials undergoing very large deformations can be modeled successfully [56]. The Lagrangian RKPM formulation has therefore been used to model problems involving plasticity [56], structural dynamics [137], and hyperelasticity [58][44]. However, Lagrangian RKPM breaks down when the mapping between the deformed and the undeformed configuration is no longer one-to-one and the deformation gradient is no longer invertible. This is the case when modeling extremely large deformation problems, such as high velocity impact and penetration processes, during which new surface formation and material fragmentation occur.

To overcome this limitation, the semi-Lagrangian RKPM formulation [55][97] was developed, where the approxi mation functions are constructed directly in the current deformed configuration, meaning that the mapping between the undeformed and the deformed configuration is no longer required. Semi-Lagrangian RKPM has been used for a large variety of problems, such as impact problems involving earth moving simulations [97], fragmentation [63][98], and slope stability analyses [89]. A downside of the semi-Lagrangian RK approximation is, however, its high computational cost. In fact, in order to reconstruct the RK approximation in the current configuration, the search for node neighbors of each evaluation point and the reconstruction of semi-Lagrangian shape functions need to be performed at each time step. This makes the semi-Lagrangian RK approach computationally expensive. On the contrary, the Lagrangian RK approach is computationally advantageous, given that its approximation is only built in the reference configuration.

Over the years, as highlighted in [127], several approaches have been proposed to couple the finite element method and meshfree approaches. These include, among others, master-slave couplings [119][123][38][39], where meshfree particles adjacent to the finite element domain are rigidly connected to finite elements, bridging domain couplings [127], where the coupling is performed by blending the two approximations at the energy level by means of a ramp function, couplings via Lagrange multipliers [10], and couplings via mixed interpolation [121]. In [121] coupling between FEM and element-free Galerkin (EFG) approximations is achieved by the definition of a transition zone (having the width of an element), where a mixed approximation was built by substituting finite element nodes with particles and by blending the FEM and EFG approximations via ramp functions so that continuity and first-order consistency are preserved. A similar approach was employed in [70] to blend EFG and enrichment functions for crack tip fields.

In [1] and in [136], a blending between FEM and meshfree approaches (EFG and RKPM, respectively) where arbitrary order reproducing conditions are imposed in the transition zone. Since the reproducing conditions are imposed to build the mixed approximation in the transitioning region, this approach does not require the definition of a ramp function. Imposing the reproducing conditions in the transitioning zone, however, requires knowledge of the FEM shape function values at the evaluation points of interest. They are, in fact, used as input when the reproducing conditions are imposed to construct the meshfree functions.

2.2 The peridynamic theory

Peridynamics is an integro-differential nonlocal reformulation of the classical theory of continuum mechanics. It was introduced in [111][113] to model spontaneous formation of cracks. Since balance laws are computed through integration rather than differentiation, assumptions on the spatial differentiability of displacement fields are not required. For this reason, the peridynamic formulation remains valid in the presence of displacement discontinuities and is thus directly applicable to problems involving material failure and damage. Peridynamics has therefore been used to model, among others, crack nucleation [14][114], crack propagation and growth in brittle glass [6][7][147][148], and failure and damage in concrete [13][133], composites [9] [18][19][143][135][144][149], and polycrystals [15][87][71]. Being nonlocal, peridynamic models possess a length scale, called horizon, which defines the range of nonlocal interactions. In the limit of vanishing nonlocality, relationships between peridynamic and classical models can be established, if the latter are well defined. In [115] it is shown that, under suitable smoothness assumptions, elastic peridynamic models reduce to classical elastic models as the horizon value approaches zero. Since governing equations in peridynamics are continuous, different discretization schemes have been employed and studied for their solution, including finite difference schemes [141], the finite element method [7][16][27][28][42][100][106][140][141], and direct quadrature and particle-based methods [16][25][85][112]. Connections between node-based discretizations of strong forms of peridynamic equations and discretized classical partial differential equations (PDEs) have been explored in [37][83][96]. In [96] the equivalence between nearest-neighbor peridynamic discretizations and finite difference schemes is shown when appropriate integration weights are chosen in the peridynamic discretization. A connection between approximations of classical continuum models with the Reproducing Kernel Particle Method (RKPM) [56][138] and peridynamics has been made in [83] for uniform discretizations, while conditions for the equivalence of Smoothed Particle Hydrodynamics (SPH) [54] and node-based peridynamics were introduced in [37].

In this work, we focus on meshfree type discretizations of the nonlocal peridynamic equations. The accuracy of the numerical solutions obtained by a node-based discretization of nonlocal peridynamic problems and their convergence to the nonlocal analytical solutions under grid refinement have been investigated in [93][94][117]. In [117] the authors solved nonlocal elliptic boundary value problems. They observed an oscillatory behavior in the convergence results and attributed it mostly to the integration error associated with the node-based integration scheme, which was implemented using the so-called full volume approach [112]. Further convergence studies were carried out in [93], where several schemes to improve the integration accuracy were employed. While improving the integration accuracy reduced the oscillatory behavior, it was noted that the attainable convergence rate was limited to first order due to a piecewise-constant approximation associated with the node-based discretization. Higher convergence rates can be obtained by employing higher-order discretizations. This can be achieved, for example, by piecewise linear finite element discretizations [140][141]. However, the use of finite element discretizations for solving peridynamic problems involving fractures can become cumbersome due to the need of adapting the mesh to the evolving cracks. Furthermore, the use of the finite element method in peridynamics can be computationally expensive as it is based on weak forms, which lead to the need of performing double spatial integration. Lastly, in the finite element framework, the order of continuity and completeness of the approximate solution are intertwined.

2.3 Waveform relaxation

The Waveform Relaxation method (WR), also called dynamic iteration method, is an iterative technique for solving space-time problems and can be used for solving the related systems of ODEs. It offers so-called parallelization of time and due to its implicit nature provides improved stability compared to explicit methods [41].

As highlighted in [35], time parallel methods can be grouped in different categories: methods based on multiple shooting [51, 2, 91], direct time parallel methods [139, 11, 3, 107] (mainly based on predictor-corrector type of approaches), space-time multigrid techniques [134][32] and waveform relaxation methods [68, 17, 76, 84]. The multiple shooting method was developed in its discrete form in [2] and then expanded in continuous form in [91]. The idea behind this approach is to partition the time domain into windows and, after a first coarse serial solution used to determine approximate initial conditions for each time window, solve in parallel a refined solution over each window. The solution at the final time step for a window might not coincide with the assumed initial condition of the next one (thus the term *shooting*): the solution is therefore performed iteratively until agreement is achieved. Different from the multiple shooting methods, multigrid methods in space and time are not naturally parallel [35], but they can be used to solve systems simultaneously over the entire space-time domain [35][31]. Following a similar idea where space-time is one large domain, a common finite element discretization of both the space and the time domains was proposed in [33].

Here, we focus on the class of time-parallel Waveform Relaxation methods [68][52]. The WR method was mostly applied to parabolic initial value problems with applications in electrical network systems and the analysis of circuits [68] and has rarely been applied to hyperbolic problems that arise in structural dynamics.

This method differs from most standard iterative techniques in that it is a continuous-time method, iterating with functions in time, called waveforms. In other words, the space domain in this algorithm is partitioned into smaller subsystems while time is not discretized but rather kept continuous. However, in practical implementations of the method, discretization of time is also desired, especially when non-linearities are considered. One of the advantages of the WR schemes is it being unstructured in the time domain. Hence, if in a system there are several variables changing at different rates, they can be uncoupled and each integrated with the appropriate time step size, with a predefined interpolating operator to exchange the information between subsystems [41]. Several ways for accelerating the convergence, thus improving the efficiency, of the waveform relaxation schemes have been investigated, such as successive over relaxation (SOR), Polynomial acceleration (Chebyshev iteration, Krylov subspace acceleration) and Multigrid [52, 48, 47, 109, 131, 110].

The SOR waveform relaxation and its convergence were first studied by Miekkala and Nevanlinna [129][130]. Reichelt et al. [82] improved the results by proposing a convolution SOR acceleration of the waveform relaxation scheme in which, instead of multiplying by a fixed SOR parameter as in the traditional SOR approach, a time-dependent SOR kernel was used. The effectiveness of this approach was confirmed in [49], where the performance of the convolution SOR was compared to SOR methods based on matrix splitting and extrapolation. It was found that, while the latter techniques led only to a minor acceleration, the method based on convolution achieved convergence acceleration similar to the one obtained for optimal stationary SOR methods. Lastly, a parareal Schwarz waveform relaxation algorithm was recently proposed by combining the aforementioned multiple shooting methods and the Schwarz waveform relaxation method [35][76][50][146]. In this scheme, similar to the space-time multigrid methods, simultaneous iteration over a set of unknowns in the space-time domain is performed [35]. The focus of these studies and procedures has been mainly on parabolic first order differential equations; limited studies have been conducted on hyperbolic and second-order systems. In [145] and [67] the application of waveform relaxation methods to second-order differential equations and their convergence have been studied; however this was performed by converting the system to first order. None of these studies considered the Newmark family of integration methods for second order ODEs that is common in structural dynamics.

Acknowledgments

A portion of this chapter has been published in Marco Pasetto, Haim Waisman, J.S. Chen, “A waveform relaxation Newmark method for structural dynamics problems”, *Computational Mechanics*, 63(6), pp. 1223-1242, 2019. The dissertation author was the primary investigator of this material.

A portion of this chapter has been published in M. Pasetto, Y. Leng, J.S. Chen, J.T. Foster, P. Seleson, “A reproducing kernel enhanced approach for peridynamic solutions”, *Computer Methods in Applied Mechanics and Engineering*, 340, 2018. The dissertation author was the primary investigator of the included material.

A portion of this chapter is currently being prepared for submission for publication of the material. M. Pasetto, J. Baek, J.S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, “A Lagrangian/semi-Lagrangian Coupling Approach for Meshfree Modeling of Extremely Large Deformation Problems”. The dissertation author was the primary investigator of this material.

Chapter 3

The reproducing kernel particle method formulation

3.1 Lagrangian reproducing kernel approximation

The Reproducing Kernel (RK) approximation is the basis of the RKPM [56][138]. The RKPM is originally based on a continuous corrected kernel estimate of a scalar function $f(\mathbf{x})$ denoted by $f^h(\mathbf{x})$:

$$f(\mathbf{x}) \approx f^h(\mathbf{x}) = \int_{\Omega} \Psi(\mathbf{x};\mathbf{x}-\mathbf{y}) f(\mathbf{y}) d\mathbf{y}, \tag{3.1}$$

where $\Omega \subset \mathbb{R}^d$ denotes a closed domain, $\mathbf{x} \in \Omega$, and

$$\Psi(\mathbf{x};\mathbf{x}-\mathbf{y}) = \Phi_a(\mathbf{x}-\mathbf{y}) C(\mathbf{x};\mathbf{x}-\mathbf{y}), \tag{3.2}$$

where $\Phi_a(\mathbf{x}-\mathbf{y})$ is a kernel function with compact support measure a used to define the locality and the order of continuity, and $C(\mathbf{x};\mathbf{x}-\mathbf{y})$ is a correction function used to impose the reproducing conditions for desired $\overline{\Omega}$ order of completeness of the approximation. Eq. (3.1) is then discretized by means of numerical integration [56]. To do so, integration weights need to be defined to ensure completeness in the approximation. Furthermore, it is shown in [56] that in order for the reproducing conditions to remain satisfied in the discrete setting, the same integration scheme needs to be used consistently throughout the whole approximation construction procedure. To maintain consistency in the discrete approximation and avoid complexity in defining the weights in the domain of integration of Eq. (3.1), an RK approximation developed directly in a discrete form was introduced in [57].

In mesh-based approaches such as the finite element method (FEM), the shape functions and the related approximation space are strictly linked to element connectivity. In the RKPM, instead, the approximate solutions are constructed over a nodal discretization of the considered domain. Conditions for particle (node) distributions to be suitable for numerical computations are outlined in [108].

Let us now consider the initial reference domain $\Omega_X \subset \mathbb{R}^d$, which after deformation gets mapped to the deformed domain $\Omega_x \subset \mathbb{R}^d$. Let Ω_X be discretized by a set of NP nodes $\{\mathbf{X}_I | \mathbf{X}_I \in \Omega_X\}_{I=1}^{NP}$, where $\mathbf{X}$ is the material coordinate defined in the reference (undeformed) domain configuration. In a Lagrangian reproducing kernel approximation, the discrete RK approximation $f^{h,L}(\mathbf{X},t)$ of a generic function $f(\mathbf{X},t)$ in Ω_X is defined as:

$$f^{h,L}(\mathbf{X},t) = \sum_{I=1}^{NP} \Psi_I^L(\mathbf{X}) f_I(t), \tag{3.3}$$

where $\{\Psi_I^L(\mathbf{X})\}_{I=1}^{NP}$ is the set of Lagrangian RK shape functions and $\{f_I\}_{I=1}^{NP}$ is the set of nodal coefficients of the approximation [56][57][138]. It has to be noted that the RK shape functions do not typically possess the Kronecker delta property. Therefore, the RK nodal coefficients (also called generalized coefficients) are not equivalent to the approximate function values at the nodes (i.e., $f^h(X_I,t) \neq f_I(t)$). However, the Kronecker delta property can be recovered through the use, for example, of singular kernel functions [64] or of the so-called transformation method [56][29], which modifies the shape functions to enforce the Kronecker-delta property.

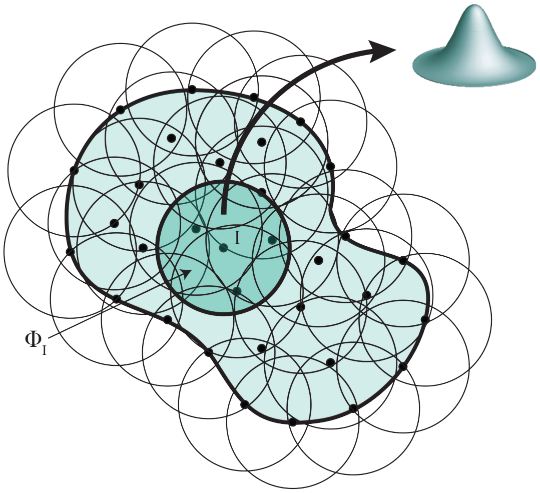


Figure 3.1: Meshfree discretization with an RK approximation kernel function.

Similarly to Eq. (3.2), RK shape functions in the discrete setting are defined as the product of a kernel function $\Phi_a(\mathbf{X}-\mathbf{X}_I)$ with compact support Φ_I (see Figure 3.1) and a correction function $C(\mathbf{X};\mathbf{X}-\mathbf{X}_I)$:

$$\Psi_I^L(\mathbf{X}) = \Phi_a(\mathbf{X}-\mathbf{X}_I)C(\mathbf{X};\mathbf{X}-\mathbf{X}_I). \tag{3.4}$$

The correction function $C(\mathbf{X};\mathbf{X}-\mathbf{X}_I)$ is defined as a linear combination of basis functions h_k, $k = 1,2,...,n$, and coefficients q_k:

$$C(\mathbf{X};\mathbf{X}-\mathbf{X}_I) = \mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\mathbf{q}(\mathbf{X}). \tag{3.5}$$

In the case of monomial basis functions:

$$C(\mathbf{X};\mathbf{X}-\mathbf{X}_I) = \sum_{|\alpha|\leq n}(\mathbf{X}-\mathbf{X}_I)^{\alpha}q_{\alpha}(\mathbf{X}) \equiv \mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\mathbf{q}(\mathbf{X}), \tag{3.6}$$

where $\mathbf{q}(\mathbf{X})$ is the column vector of coefficients and $\mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)$ is the row vector of the monomial bases:

$$\mathbf{H}^T(\mathbf{X}-\mathbf{X}_I) = \{(\mathbf{X}-\mathbf{X}_I)^{\alpha}\}_{|\alpha|\leq n}, \tag{3.7}$$

where n is the basis order and, in multi-index notation, $\alpha \equiv (\alpha_1,\alpha_2,...,\alpha_d)$, $|\alpha| \equiv \sum_{i=1}^{d}\alpha_i$, $\mathbf{X}^{\alpha} \equiv X_1^{\alpha_1}X_2^{\alpha_2}...X_d^{\alpha_d}$, and $\mathbf{X}_I^{\alpha} \equiv X_{I1}^{\alpha_1}X_{I2}^{\alpha_2}...X_{Id}^{\alpha_d}$

A unique property of the RK approximation is that, different from the FEM, the order of approximation is completely independent of the smoothness (order of continuity) of the approximation, allowing for great flexibility in the definition of the approximation function. The basis order n defines the order of completeness in the RK approximation. Smoothness of the approximation functions is determined through the choice of the kernel function $\Phi_a(\mathbf{x}-\mathbf{x}_I)$. A cubic B-spline function such as

$$\Phi_a\left(\frac{z}{a}\right) = \begin{cases} \frac{2}{3} - 4(\frac{z}{a})^2 + 4(\frac{z}{a})^3 & \text{for } 0 \leq \frac{z}{a} < \frac{1}{2} \\ \frac{4}{3} - 4(\frac{z}{a}) + 4(\frac{z}{a})^2 - \frac{4}{3}(\frac{z}{a})^3 & \text{for } \frac{1}{2} \leq \frac{z}{a} \leq 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.8}$$

where

$$z = \| \mathbf{X} - \mathbf{X}_I \|, \tag{3.9}$$

for example, gives a C^2 continuity. This kernel function Φ_a has a spherical support (see Figure 3.2 for the 2D case). The kernel function can also be constructed by multiplying the chosen kernel function in one-dimension according to:

$$\Phi_{\mathbf{a}}(\mathbf{X} - \mathbf{X}_I) = \prod_{i=1}^{d} \Phi_{a_i}(X_i - X_{Ii}), \tag{3.10}$$

where a_i represents the chosen support measure in each dimension. In this case, the kernel function $\Phi_{\mathbf{a}}$ has a brick support (see Figure 3.2 for the 2D case). In addition, in [46] RK kernel functions defined on triangulations are employed.

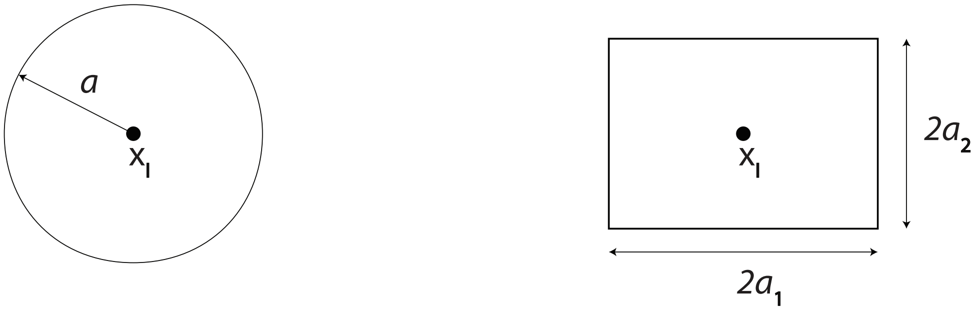


Figure 3.2: Spherical (circular in 2D) and brick (rectangular in 2D) support for Φ_a.

The coefficients $\mathbf{q}(\mathbf{X})$ in Eq. (3.6) are determined by imposing the reproducing conditions in the reference initial configuration:

$$\sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})\mathbf{H}(\mathbf{X}_I) = \mathbf{H}(\mathbf{X}), \tag{3.11}$$

that, in the case of monomial reproducing conditions becomes:

$$\sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})\mathbf{X}_I^{\alpha} = \mathbf{X}^{\alpha}, |\alpha| \leq n. \tag{3.12}$$

Equivalently, by the binomial theorem,

$$\begin{aligned}
\sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})(\mathbf{X}-\mathbf{X}_I)^{\alpha} &= \sum_{I=1}^{NP} \Psi_I(\mathbf{X}) \prod_{p=1}^{d} \sum_{k_p=0}^{\alpha_p} \binom{\alpha_p}{k_p} (-1)^{k_p} X_p^{\alpha_p-k_p} x_{Ip}^{k_p} \\
&= \sum_{I=1}^{NP} \Psi_I(\mathbf{X}) \sum_{k_1=0}^{\alpha_1} ... \sum_{k_d=0}^{\alpha_d} \binom{\alpha_1}{k_1} ... \binom{\alpha_d}{k_d} (-1)^{|k|} \mathbf{X}^{\alpha-k}\mathbf{X}_I^k \\
&= \sum_{k_1=0}^{\alpha_1} ... \sum_{k_d=0}^{\alpha_d} \binom{\alpha_1}{k_1} ... \binom{\alpha_d}{k_d} (-1)^{|k|} \mathbf{x}^{\alpha-k} \sum_{I=1}^{NP} \Psi_I(\mathbf{X})\mathbf{X}_I^k \\
&= \sum_{k_1=0}^{\alpha_1} ... \sum_{k_d=0}^{\alpha_d} \binom{\alpha_1}{k_1} ... \binom{\alpha_d}{k_d} (-1)^{|k|} \mathbf{X}^{\alpha-k}\mathbf{X}^k \\
&= \mathbf{X}^{\alpha} \sum_{k_1=0}^{\alpha_1} ... \sum_{k_d=0}^{\alpha_d} \binom{\alpha_1}{k_1} ... \binom{\alpha_d}{k_d} (-1)^{|k|} \\
&= \mathbf{X}^{\alpha}\delta_{\alpha 0} = \delta_{\alpha 0},
\end{aligned} \tag{3.13}$$

where $\delta_{\alpha 0} = \delta_{\alpha_1 0}\delta_{\alpha_2 0}...\delta_{\alpha_d 0}$, meaning that

$$\sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})(\mathbf{X}-\mathbf{X}_I)^{\alpha} = \delta_{\alpha 0}, \tag{3.14}$$

$$\sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})\mathbf{H}(\mathbf{X}-\mathbf{X}_I) = \mathbf{H}(\mathbf{0}), \tag{3.15}$$

where $\mathbf{H}(\mathbf{0})$ is a vector containing the values of $\delta_{\alpha 0}$ for all the considered values of α. By using (3.4) and (3.5) in Eq. (3.15) one obtains:

$$\sum_{I=1}^{NP} \Phi_a(\mathbf{X}-\mathbf{X}_I)[\mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\mathbf{q}(\mathbf{X})]\mathbf{H}(\mathbf{X}-\mathbf{X}_I) = \mathbf{H}(\mathbf{0}), \tag{3.16}$$

$$\left[\sum_{I=1}^{NP} \Phi_a(\mathbf{X}-\mathbf{X}_I)\mathbf{H}(\mathbf{X}-\mathbf{X}_I)\mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\right]\mathbf{q}(\mathbf{X}) = \mathbf{H}(\mathbf{0}). \tag{3.17}$$

Consequently,

$$\mathbf{q}(\mathbf{X}) = \mathbf{M}^{-1}(\mathbf{X})\mathbf{H}(\mathbf{0}), \tag{3.18}$$

where $\mathbf{M}(\mathbf{x})$ is called the moment matrix and is defined as follows:

$$\mathbf{M}(\mathbf{X}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{X}-\mathbf{X}_I)\mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\Phi_a(\mathbf{X}-\mathbf{X}_I). \tag{3.19}$$

By substituting $\mathbf{q}(\mathbf{X})$ obtained from (3.18) into (3.6), the RK shape functions in (3.4) are obtained as

$$\begin{aligned}\Psi_I^L(\mathbf{X}) &= \mathbf{H}^T(\mathbf{X}-\mathbf{X}_I)\mathbf{M}^{-1}(\mathbf{X})\mathbf{H}(\mathbf{0})\Phi_a(\mathbf{X}-\mathbf{X}_I)\\ &= \mathbf{H}^T(\mathbf{0})\mathbf{M}^{-1}(\mathbf{X})\mathbf{H}(\mathbf{X}-\mathbf{X}_I)\Phi_a(\mathbf{X}-\mathbf{X}_I).\end{aligned} \tag{3.20}$$

It can be easily shown that the expression for $\Psi_I^L(\mathbf{X})$ in (3.20) satisfies Eq. (3.12) [108].

In order for the expressions in (3.18) and (3.20) to be defined, the moment matrix in (3.19) must be invertible, meaning that the reproducing equations are required to be linearly independent at any evaluation point $\mathbf{X} \in \Omega_X$. This is possible only if there is a sufficient number n_p of independent kernels covering the evaluation point $\mathbf{X}$ [56][108], where

$$n_p = \binom{n+d}{d}. \tag{3.21}$$

Kernel independence means that the associated points are not located on a geometry that can be represented by the basis functions. For example, if linear bases are used in 3D, every evaluation point should be covered by 4 kernel supports $\Phi_I(\mathbf{X})$ of non-coplanar points. The algorithm to construct the RK shape functions is provided in the form of a pseudo-code in Appendix 3-A, while the reproducing capabilities of the RK shape functions are shown in Appendix 3-B.

3.2 Semi-Lagrangian reproducing kernel approximation

In the semi-Lagrangian approach [55][97], while the RK discretization points follow the material motion, function approximation is built by means of shape functions expressed and constructed with respect to the current spatial coordinates:

$$\tilde{f}(\varphi(\mathbf{X},t),t) = \tilde{f}(\mathbf{x},t) \approx \tilde{f}^{h,SL}(\mathbf{x},t) = \sum_{I=1}^{NP} \Psi_I^{SL}(\mathbf{x})\tilde{f}_I(t), \tag{3.22}$$

thus avoiding the need for directly knowing the mapping φ [55][97][128] between the reference and current configuration. The semi-Lagrangian shape functions are defined as

$$\Psi_I^{SL}(\mathbf{x}) = C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I), \tag{3.23}$$

where $\Psi_I^{SL}(\mathbf{x})$ is the shape function of node I built in the current configuration, with $\mathbf{x}_I = \varphi(\mathbf{X}_I,t)$, and $C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$ and $\Phi_a(\mathbf{x}-\mathbf{x}_I)$ are the correction and kernel function expressed in the current configuration, respectively. Similarly to the Lagrangian case presented in Section 3.1, the kernel function controls the smoothness and the locality of the approximation, while the correction function is defined as a linear combination of basis functions and coefficients

$$C(\mathbf{x};\mathbf{x}-\mathbf{x}_I) = \mathbf{H}^T(\mathbf{x}-\mathbf{x}_I)\mathbf{q}(\mathbf{x}), \tag{3.24}$$

where $\mathbf{H}^T(\mathbf{x}-\mathbf{x}_I)$ is the row basis vector and $\mathbf{q}(\mathbf{x})$ is the column vector of coefficients. As in the Lagrangian approximation case, the coefficients vector $\mathbf{q}(\mathbf{x})$ is determined by imposing reproducing conditions, but this time in the current deformed configuration. For example, for the case of monomial basis and monomial reproducing conditions:

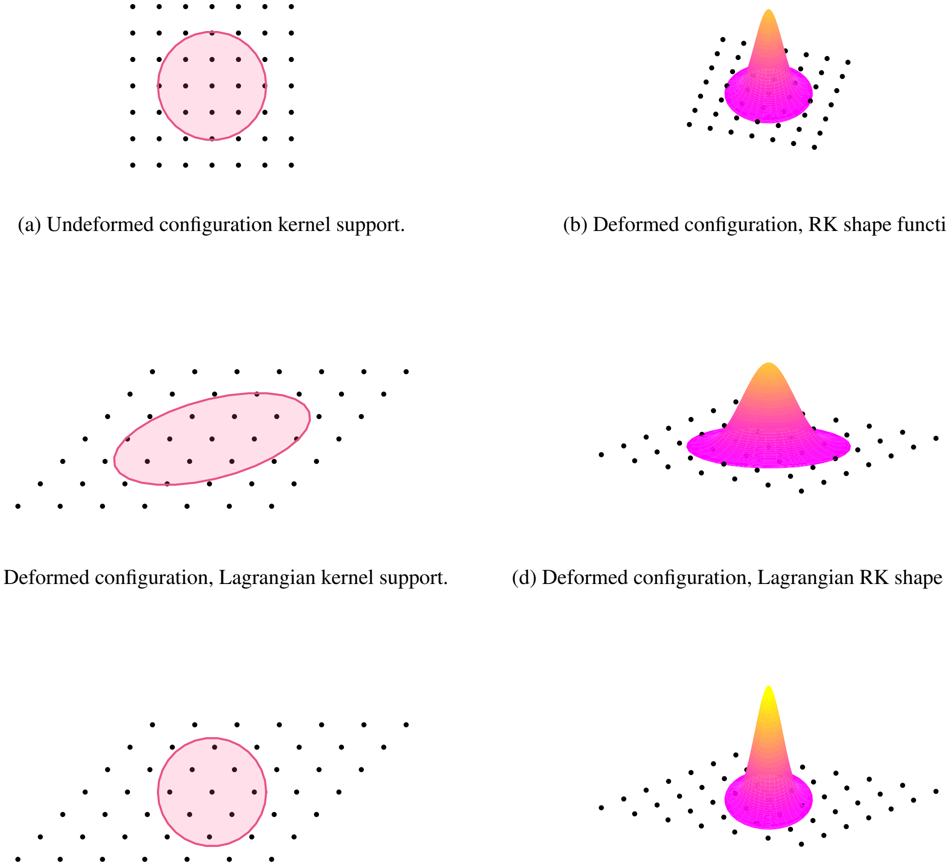
$$\sum_{I=1}^{NP} \Psi_I^{SL}(\mathbf{x})\mathbf{x}_I^{\alpha} = \mathbf{x}^{\alpha}, |\alpha| \leq n, \tag{3.25}$$

where NP, n, α, and $|\alpha|$ are defined as in Section 3.1. After following similar steps as in Section 3.1, the semi-Lagrangian shape functions can be expressed as

$$\Psi_I^{SL}(\mathbf{x}) = \mathbf{H}^T(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I), \tag{3.26}$$

where $\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x}-\mathbf{x}_I)\mathbf{H}^T(\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I)$ is the moment matrix defined in the current configuration.

As in the Lagrangian case (see Section 3.1), kernel functions can be defined either with a spherical or a brick support. In the Lagrangian RK approximation, though, the kernel functions, which are defined in the reference configuration, deform together with the material body and cover the same group of material nodes before and after deformation. The semi-Lagrangian kernels, instead, traditionally do not deform with the body and cover different RK nodes over the course of the deformation. Figure 3.3 shows the comparison between Lagrangian and semi-Lagrangian kernel supports during deformation. In order for $\mathbf{M}(\mathbf{x})$ to be invertible in Eq. (3.26), n_p (see Eq. (3.21)) independent kernels need to cover evaluation point $\mathbf{x}$ in the current configuration. However, given that the kernels cover different points at different times, it can happen that, at a certain point during the deformation, the number of neighbouring nodes covering x is not enough for the chosen basis order n. This is the case, for example, in fragmentation problems, when a single particle detaches itself from the body and has only itself as a neighbour. In order not to be restricted, in these situations, to the use of constant bases or to an abrupt reduction in basis order, a “quasi-linear” formulation was introduced in [20]. The quasi-linear reproducing kernel approach allows for almost linear completeness when the number of neighbours is insufficient to achieve first-order completeness.



(a) Undeformed configuration kernel support.

(b) Deformed configuration, RK shape function.

(c) Deformed configuration, Lagrangian kernel support.

(d) Deformed configuration, Lagrangian RK shape function.

(e) Deformed configuration, semi-Lagrangian kernel support.

(f) Deformed configuration, semi-Lagrangian RK shape function.

Figure 3.3: Lagrangian and semi-Lagrangian kernels and associated RK shape functions. The shape functions are built using a linear basis and a cubic B-spline kernel.

It has to be noted that, differently from the Lagrangian case, the material time derivative of the semi-Lagrangian shape functions does not vanish. This is due to their time dependency and to their construction in the current configuration, which leads to the kernel supports covering different material particles at different times (see Figure 3.3). The temporal derivative $\dot{\tilde{f}}^{h,SL}(\mathbf{X},t)$

of the semi-Lagrangian approximation of a function $\tilde{f}^{h,SL}(\mathbf{X},t)$ is

$$\dot{\tilde{f}}^{h,SL}(\mathbf{X},t) = \sum_{I=1}^{NP}[\Psi_I^{SL}(\mathbf{x})\dot{\tilde{f}}_I(t) + \Psi_I^{*}(\mathbf{x})\tilde{f}_I(t)], \tag{3.27}$$

where $\dot{\tilde{f}}_I(t)$ is the I-th nodal coefficient corresponding to $\dot{\tilde{f}}^{h,SL}(\mathbf{x},t)$ and $\Psi_I^{*}(\mathbf{x})$ is the correction due to the time rate of the semi-Lagrangian kernel $\dot{\Phi}_a$ [98]:

$$\Psi_I^{*}(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}_I - \mathbf{x})\dot{\Phi}_a(\mathbf{x} - \mathbf{x}_I). \tag{3.28}$$

As shown in Section 3.1 for the Lagrangian case, the kernel Φ_a is typically a function of a distance measure normalized with respect to support size measures (a for the spherical support case and a_i, $i = 1,2,3$, in the case of a brick support). Therefore, for the case, for example, with a spherical support:

$$\dot{\Phi}_a(\mathbf{x} - \mathbf{x}_I) = \left.\frac{\partial \Phi_a}{\partial t}\right|_{[\mathbf{X}]} = \Phi_a'\left(\frac{\| \mathbf{x} - \mathbf{x}_I \|}{a}\right)\frac{\mathbf{x} - \mathbf{x}_I}{a \| \mathbf{x} - \mathbf{x}_I \|} \cdot \frac{\partial(\mathbf{x} - \mathbf{x}_I)}{\partial t}. \tag{3.29}$$

A similar expression can be derived for kernels with brick support [128]. Furthermore, it has to be noted that the time derivative of the correction function is omitted in Eq. (3.28) as it is constructed in the current configuration in order to meet the reproducing conditions (see Eq. 3.25).

It is noted in [128] and [120] that the contribution of the terms involving Ψ^{*} in Eq. (3.27) is negligible for problems with moderate relative changes in nodal velocities [128] or when a nodal integration scheme, such as direct nodal integration (DNI), stabilized conforming nodal integration (SCNI), or stabilized nonconforming nodal integration (SNNI), is employed [120]. In such cases, the term associated with Ψ^{*} can be omitted.

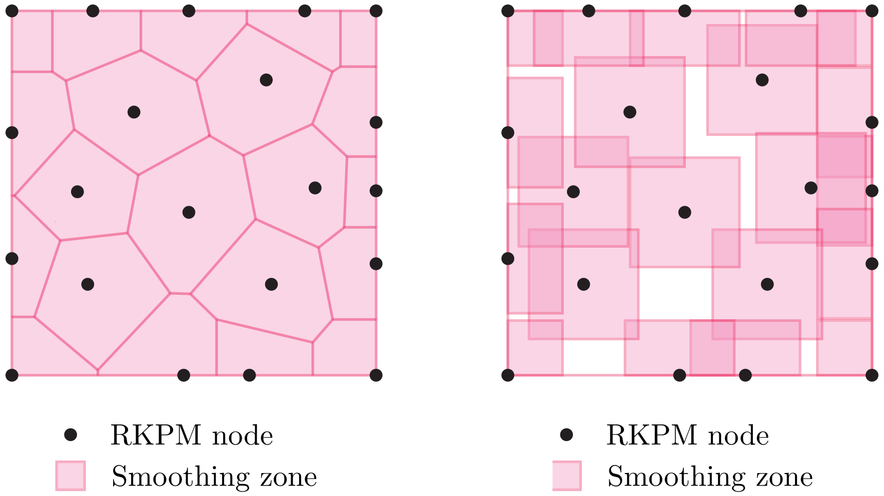
When it was introduced [55], the semi-Lagrangian approximation (see Eq. (3.22)) was employed to approximate the velocity field. The related acceleration field was determined by Eq. (3.27) but an approximation for the displacement field was not defined. In [97][63], displacement and velocity are both approximated by using Eq. (3.22), while in [128] Eq. (3.22) is used for the displacement field and velocity and acceleration are approximated by using the first and second temporal derivative of the displacement approximation, respectively. If the terms associated with the temporal derivatives of Φ_a are omitted, the two approaches coincide.

SCNI was developed to obtain optimal convergence for linearly complete functions as well as to solve the rank instability issue present in direct nodal integration [59]. In SCNI, the domain is partitioned in conforming cells (see Figure 3.4a for a two-dimensional example) each associated with an RKPM node. In order to avoid the rank instability arising from the direct evaluation of the shape functions gradients at the nodes, nodal gradients are computed through a smoothing operation over the conforming nodal domains as follows:

$$\tilde{\nabla}\Psi_I(\mathbf{x}_I) = \frac{1}{W_L}\int_{\Omega_L} \nabla\Psi_I d\Omega = \frac{1}{W_L}\int_{\Gamma_L} \Psi_I \mathbf{n} d\Gamma, \tag{3.30}$$

where $\tilde{\nabla}(\cdot)$, W_L, and $\mathbf{n}$ are a smoothing gradient operator, the volume of a gradient smoothing cell of node L, and the surface normal vector, respectively. Due to the way the smoothing is performed and to the conforming nature of the nodal cells, the first-order variational consistency for Galerkin linear exactness is satisfied, leading to optimal convergence. However, in order for SCNI to be employed in a semi-Lagrangian RKPM formulation, the conforming nodal cells would have to be reconstructed at each integration time step. For this reason, this integration approach is too computationally expensive to be feasibly used for practical problems in a semi-Lagrangian framework. Therefore, instead of SCNI, SNNI is often used in conjunction with semi-Lagrangian RK [98]. In SNNI, the smoothing of the shape function gradients in Eq. (3.30) is performed over nonconforming cells, represented by brick domains constructed around each RKPM node (see Figure 3.4b for a two-dimensional example). These smoothing domains traditionally maintain their shape, similarly to the semi-Lagrangian kernel function supports, for the whole duration of the simulation, making them suitable for semi-Lagrangian RKPM. The relaxation of the conforming requirement, however, leads to the first-order variational consistency for Galerkin linear exactness being no longer satisfied. In order to improve the performance of

SNNI and achieve an arbitrary-order variational consistency, a variationally consistent integration for meshfree methods was proposed in [61]. Furthermore, in [65] additional stabilization terms were introduced for SCNI by means of a modified strain smoothing expression (M-SCNI). In [74], instead, a naturally stabilized nodal integration (NSNI) was introduced. NSNI adds stabilization to the direct nodal integration (DNI) scheme by performing a Taylor expansion about each RKPM node of the displacement field in each related nodal integration cell.



(a) Conforming SCNI cells.

(b) Nonconforming SNNI cells.

Figure 3.4: RKPM smoothing cells for SCNI and SNNI (two-dimensional case).

Acknowledgments

A portion of this chapter has been published in M. Pasetto, Y. Leng, J.S. Chen, J.T. Foster, P. Seleson, “A reproducing kernel enhanced approach for peridynamic solutions”, *Computer Methods in Applied Mechanics and Engineering*, 340, 2018. The dissertation author was the primary investigator of the included material.

A portion of this chapter is currently being prepared for submission for publication of the material. M. Pasetto, J. Baek, J.S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, “A Lagrangian/semi-Lagrangian Coupling Approach for Meshfree Modeling of Extremely Large Deformation Prob-

lems”. The dissertation author was the primary investigator of this material.

3.3 Appendix 3-A: Pseudo-code for constructing RK shape functions

Variables:

n_{nodes}: number of RK nodes $\mathbf{x}_I$

n_{eval}: number of evaluation points $\mathbf{x}$

n: basis order

d: number of spatial dimensions

$sb = \frac{(n+d)!}{n!d!}$

a_d: kernel support measure(s)

$$\overline{\mathbf{X}}_e^d = \begin{bmatrix} x_1^1 & ... & x_d^1 \\ : & ... & : \\ x_1^i & ... & x_d^i \\ : & ... & : \\ x_1^{n_{eval}} & ... & x_d^{n_{eval}} \end{bmatrix} \text{: matrix of coordinates of all the evaluation points}$$

$$\overline{\mathbf{X}}_J^d = \begin{bmatrix} x_{1J}^1 & ... & x_{1J}^d \\ : & ... & : \\ x_{1J}^i & ... & x_{dJ}^i \\ : & ... & : \\ x_{1J}^{n_{nodes}} & ... & x_{dJ}^{n_{nodes}} \end{bmatrix} \text{: matrix of coordinates of all the RK nodes}$$

$\mathbf{x}^i = \overline{\mathbf{X}}_e^d(i,:) = [x_1^i, ..., x_d^i]$: i-th evaluation point coordinates

$\mathbf{x}_J^i = \overline{\mathbf{X}}_J^d(i,:) = [x_{1J}^i, ..., x_{dJ}^i]$: i-th RK node coordinates

$\mathbf{M}(\mathbf{x}) = \mathbf{M}_{sb\times sb}$: moment matrix

$\mathbf{S}_{n_{eval}\times n_{nodes}}$: matrix to store the RK shape function values at the evaluation points

$\mathbf{H}^T(\mathbf{0}) = \mathbf{H}_0^T = [1,0,...,0]_{1\times sb}$

$\mathbf{H}(\mathbf{x}-\mathbf{x}_J) = \mathbf{H}_{sb\times 1}$: basis vector

$\Psi_J(\mathbf{x}) = \Psi_J$: shape functions of node J evaluated at $\mathbf{x}$

Algorithm 1 Cubic B-spline Kernel (spherical support)

```
function $[\Phi_a] = Kernel(\mathbf{x}, \mathbf{x}_J, a_d)$
{%} N.B. different types of kernels can be implemented. Similarly, for the other algorithms.
$z = \frac{\|\mathbf{x}-\mathbf{x}_J\|}{a_d}$
if $0 \le z < \frac{1}{2}$ then
    $\Phi_a = \frac{2}{3} - 4z^2 + 4z^3$
else
    if $\frac{1}{2} \le z < 1$ then
        $\Phi_a = \frac{4}{3} - 4z + 4z^2 - \frac{4}{3}z^3$
    end if
    if $z \ge 1$ then
        $\Phi_a = 0$
    end if
end if
end function
```

Algorithm 2 Basis vector computation

```
function $[\mathbf{H}] = Basis(\mathbf{x}, \mathbf{x}_J, n)$
if d=1 then
    $x_1 = \mathbf{x}(1)$
    $x_{1J} = \mathbf{x}_J(1)$
    for $k = 0 : n$ do
        $\mathbf{H}(k+1) = (x_1 - x_{1J})^k$
    end for
else
    if d=2 then
        $x_1 = \mathbf{x}(1); x_2 = \mathbf{x}(2)$
        $x_{1J} = \mathbf{x}_J(1); x_{2J} = \mathbf{x}_J(2)$
        $count = 1$
        for $k = 0 : n$ do
            for $j = 0 : k$ do
                $i = k - j$
                $\mathbf{H}(count) = (x_1 - x_{1J})^i (x_2 - x_{2J})^j$
                $count = count + 1$
            end for
        end for
    end if
    if d=3 then
        $x_1 = \mathbf{x}(1); x_2 = \mathbf{x}(2); x_3 = \mathbf{x}(3)$
        $x_{1J} = \mathbf{x}_J(1); x_{2J} = \mathbf{x}_J(2); x_{3J} = \mathbf{x}_J(3)$
        $count = 1$
        for $k = 0 : n$ do
            for $j = 0 : k$ do
                for $i = 0 : k - j$ do
                    $l = k - j - i$
                    $\mathbf{H}(count) = (x_1 - x_{1J})^i (x_2 - x_{2J})^j (x_3 - x_{3J})^l$
                    $count = count + 1$
                end for
            end for
        end for
    end if
end if
end function
```

Algorithm 3 Moment matrix computation

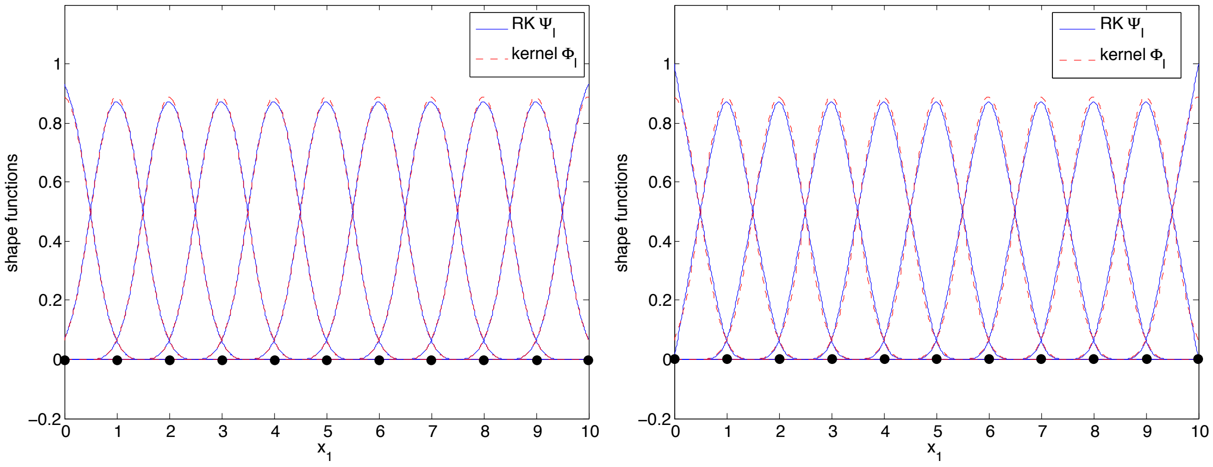
```
**function** $[\mathbf{M}] = Moment(\mathbf{x}, \overline{\mathbf{X}}_J^d, n)$
**$\mathbf{M} = \mathbf{0}$**                                        ▷ initialize
**for** $j = 1 : n_{nodes}$ **do**
    $\mathbf{x}_J = \overline{\mathbf{X}}_J^d(j, :)$
    $[\Phi_a] = Kernel(\mathbf{x}, \mathbf{x}_J, a_d)$
    $[\mathbf{H}] = Basis(\mathbf{x}, \mathbf{x}_J, n)$
    $\mathbf{M} = \mathbf{M} + \mathbf{H}\mathbf{H}^T \Phi_a$
**end for**
**end function**
```

Algorithm 4 RK shape functions $\Psi_J(\mathbf{x})$ evaluation

```
**for** $i = 1 : n_{eval}$ **do**
    $\mathbf{x} = \overline{\mathbf{X}}_e^d(i, :)$                       ▷ evaluation point coordinates
    $[\mathbf{M}] = Moment(\mathbf{x}, \overline{\mathbf{X}}_J^d, n)$
    **for** $j = 1 : n_{nodes}$ **do**
        $\mathbf{x}_J = \overline{\mathbf{X}}_J^d(j, :)$                ▷ RK node coordinates
        $[\Phi_a] = Kernel(\mathbf{x}, \mathbf{x}_J, a_d)$
        $[\mathbf{H}] = Basis(\mathbf{x}, \mathbf{x}_J, n)$
        $\Psi_J = \mathbf{H}_0^T \mathbf{M}^{-1} \mathbf{H} \Phi_a$
        $\mathbf{S}(i, j) = \Psi_J(\mathbf{x})$
    **end for**
**end for**
```

3.4 Appendix 3-B: Comparison between RK shape functions and uncorrected kernels

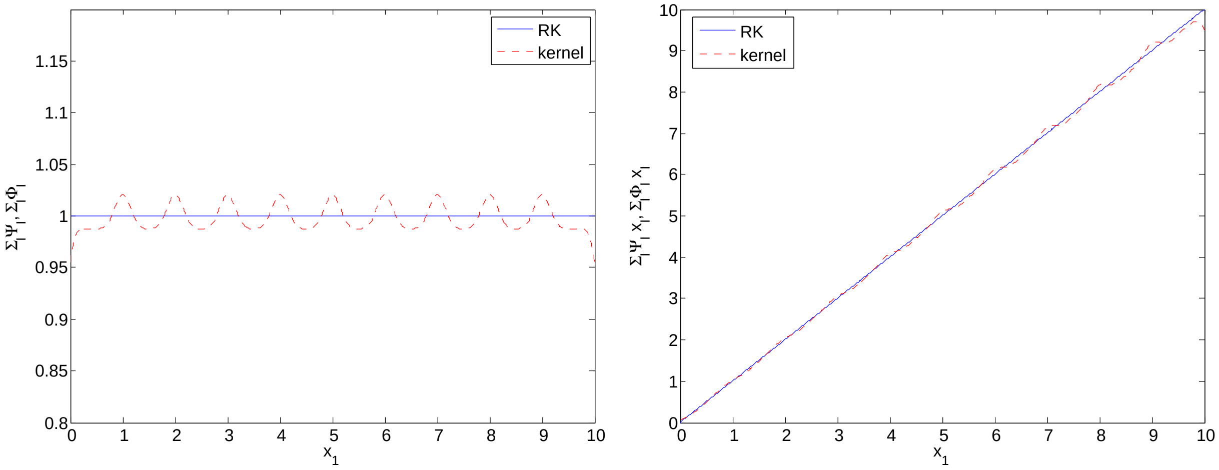
In this appendix, the reproducing capabilities of the RK shape functions are compared with those of the uncorrected kernels. Figure 3.5 shows the comparison between the one-dimensional uncorrected kernel functions $\Phi_a(x-x_I)$ and the RK shape functions $\Psi_I(x) = \Phi_a(x-x_I)C(x;x-x_I)$. A set of 11 equally spaced nodes and a cubic B-spline kernel with measure $a = 1.5h$, where h represents the nodal spacing, were used. For plotting purposes, the cubic B-spline was scaled by a factor of $\frac{4}{3}$ with respect to the one presented in Eq. (3.8). It can be noted that due to the correction function, which modifies the kernel functions especially near the boundaries, the RK approximation shape functions, with constant and linear bases (see Figures 3.5a and 3.5b, respectively) exactly reproduce constant and linear functions (see Figures 3.6a and 3.6b, respectively). On the contrary, the kernel functions do not satisfy the consistency conditions and thus cannot reproduce any order of polynomials. Similar results can be obtained for non-uniform nodal discretizations (see Figures 3.7 and 3.8).



(a) Constant basis

(b) Linear basis

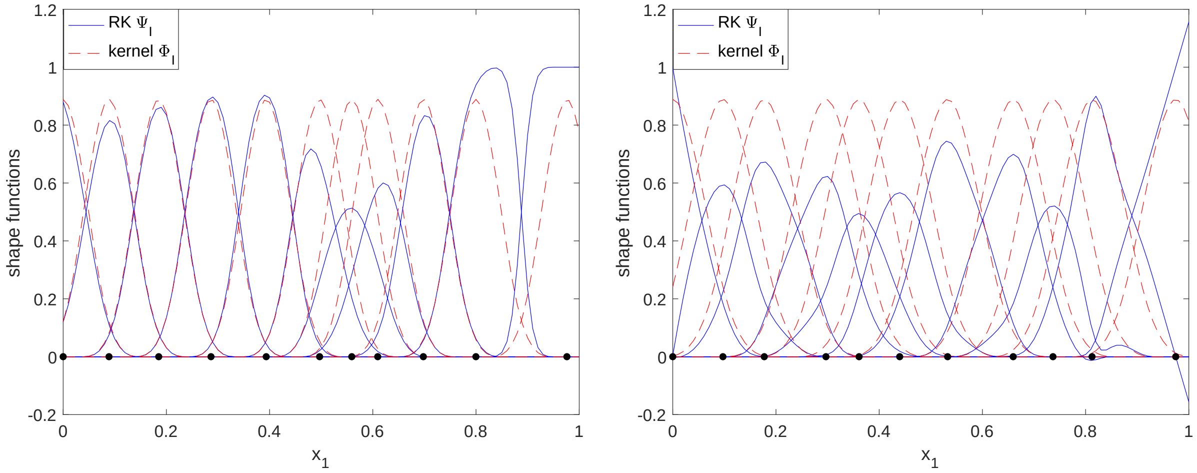
Figure 3.5: Kernel functions Φ_I and RK shape functions Ψ_I (1D) for a uniform point distribution.



(a) Zeroth-order consistency

(b) First-order consistency

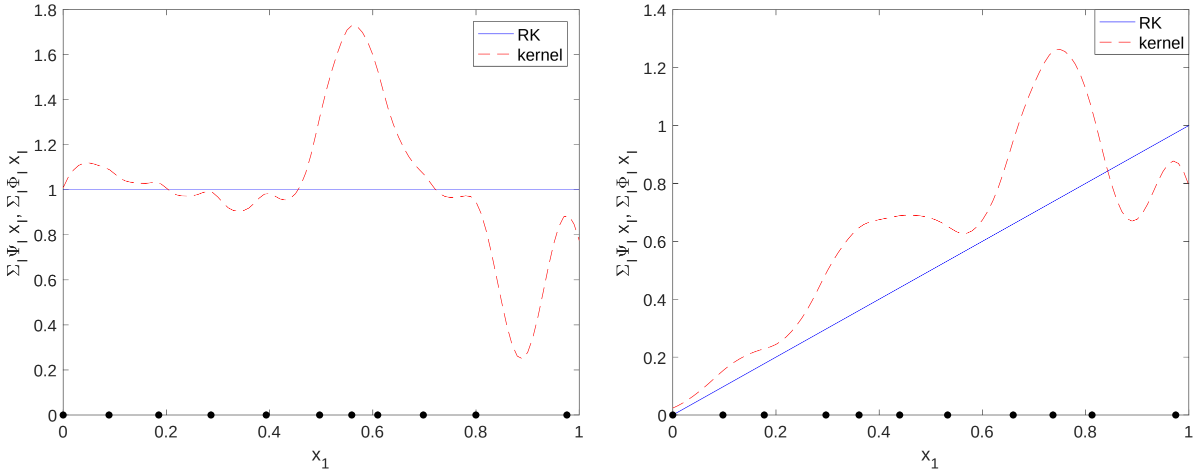
Figure 3.6: Zeroth- and first-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (1D) for a uniform point distribution. Constant and linear bases were used in the RK approximation in (a) and (b), respectively.



(a) Constant basis

(b) Linear basis

Figure 3.7: Kernel functions Φ_I and RK shape functions Ψ_I (1D) for a non-uniform point distribution.



(a) Zeroth-order consistency

(b) First-order consistency

Figure 3.8: Zeroth- and first-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (1D) for a non-uniform point distribution. Constant and linear bases were used in the RK approximation in (a) and (b), respectively.

A similar comparison is also shown for a two-dimensional case (see Figure 3.9). Figures 3.10 and 3.11 show the kernels and the RK shape functions associated with a node $\mathbf{x}_I$ near the boundary and in the interior of the domain, respectively. In this construction, a uniform grid of 5x5 nodes, constant bases, and a spherical cubic B-spline (Eq. (3.8)) with support measure $a = 1.5h$, with h being the grid spacing, were used. Similar to the one-dimensional case, the significance of the correction functions for nodes near the boundary of the domain can be observed. Satisfaction of the zeroth-order and first-order reproducing conditions by the RK approximation are shown in Figures 3.12 and 3.13, respectively. For the linear case, so as to guarantee the minimum necessary number of independent kernels, the kernel support measure was increased to $a = 2h$ and linear bases were used to construct the RK approximation functions.

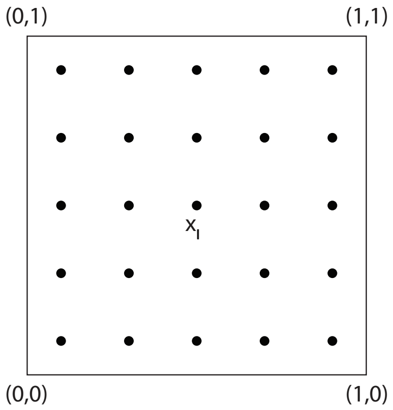
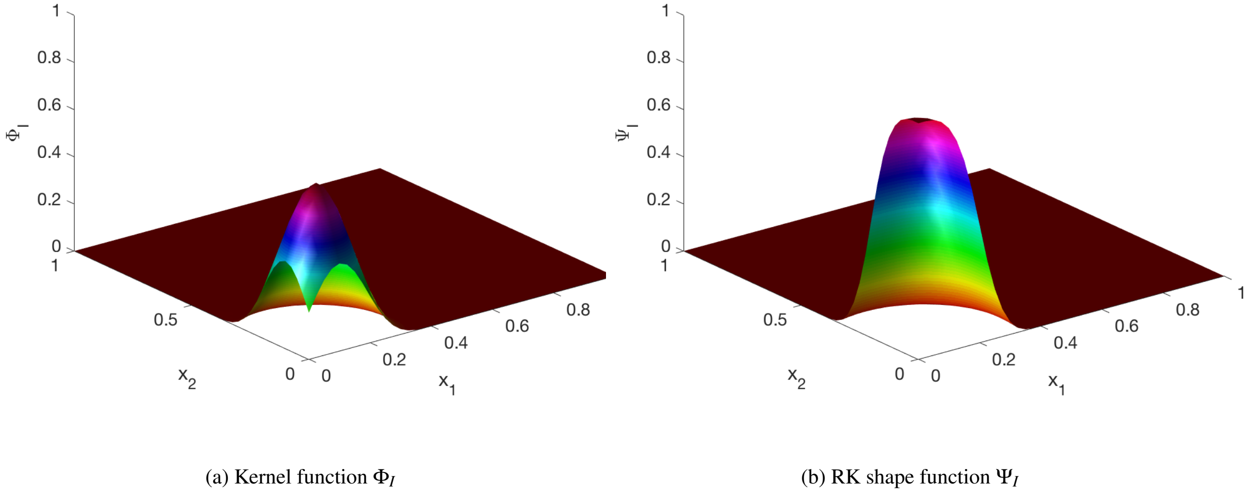


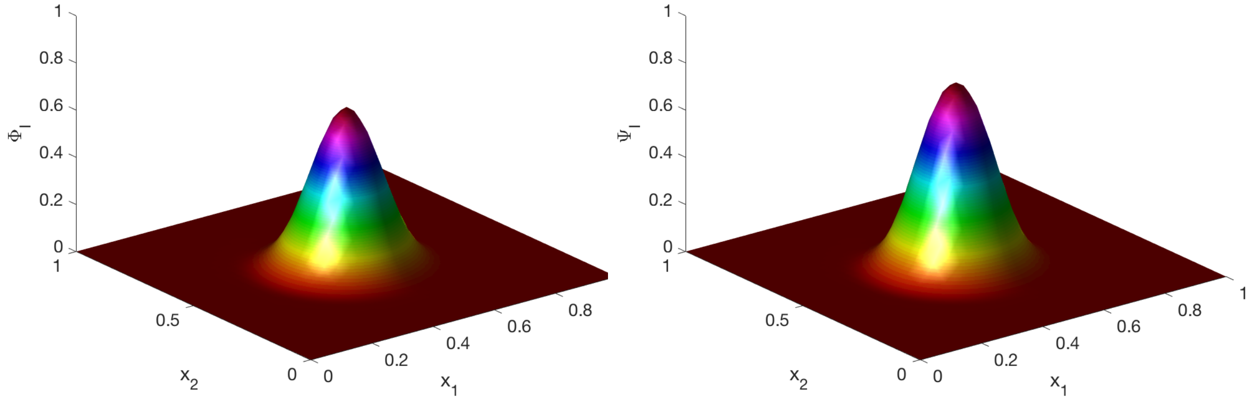
Figure 3.9: A meshfree discretization of a two-dimensional unit square domain.



(a) Kernel function Φ_I

(b) RK shape function Ψ_I

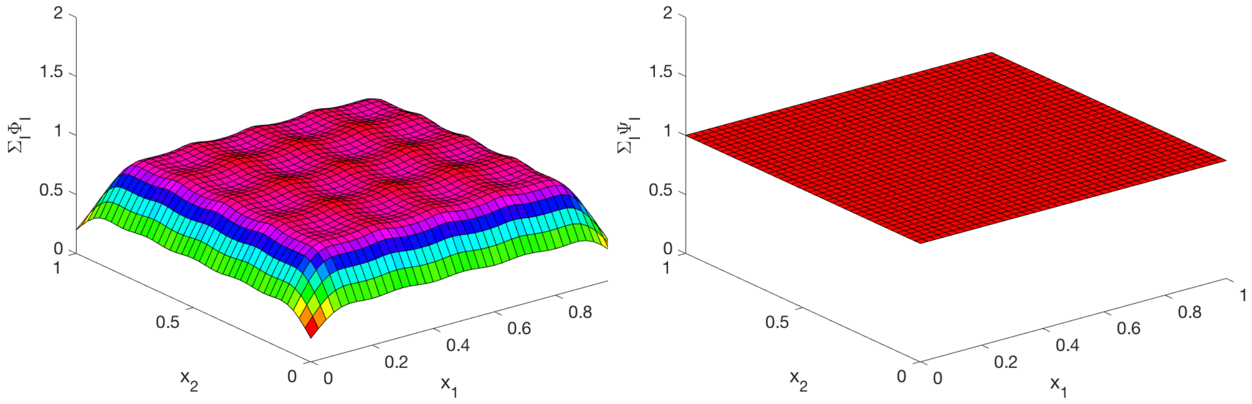
Figure 3.10: Kernel function Φ_I and RK shape function Ψ_I (2D) near the domain boundary.



(a) Kernel function Φ_I

(b) RK shape function Ψ_I

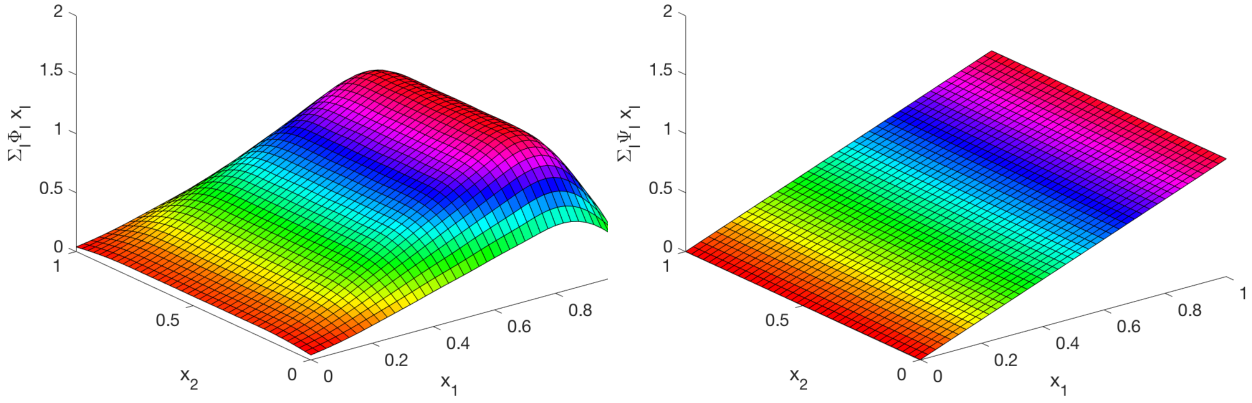
Figure 3.11: Kernel function Φ_I and RK shape function Ψ_I (2D) in the domain interior.



(a) Kernel functions Φ_I

(b) RK shape functions Ψ_I

Figure 3.12: Zeroth-order consistency conditions for the kernel functions Φ_I and RK shape functions Ψ_I (2D).



(a) Kernel functions Φ_I

(b) RK shape functions Ψ_I

Figure 3.13: First-order consistency conditions for the kernel functions Φ_I and RK Ψ_I shape functions (2D).

Chapter 4

A Lagrangian/semi-Lagrangian coupling approach for meshfree modeling

4.1 Lagrangian equation of motion

In solid mechanics, total Lagrangian and updated Lagrangian formulations are commonly used. In both formulations, the material motion from the reference (initial) domain $\Omega_X \subset \mathbb{R}^d$ to the deformed (current) domain $\Omega_x \subset \mathbb{R}^d$ is described by the mapping function $\mathbf{x} = \varphi(\mathbf{X},t)$, where $\mathbf{X}$ and $\mathbf{x}$ represent the material reference coordinates and the current spatial coordinates, respectively. The material displacement is represented by $\mathbf{u}(\mathbf{X},t) = \varphi(\mathbf{X},t) - \mathbf{X}$ (see Figure 4.1).

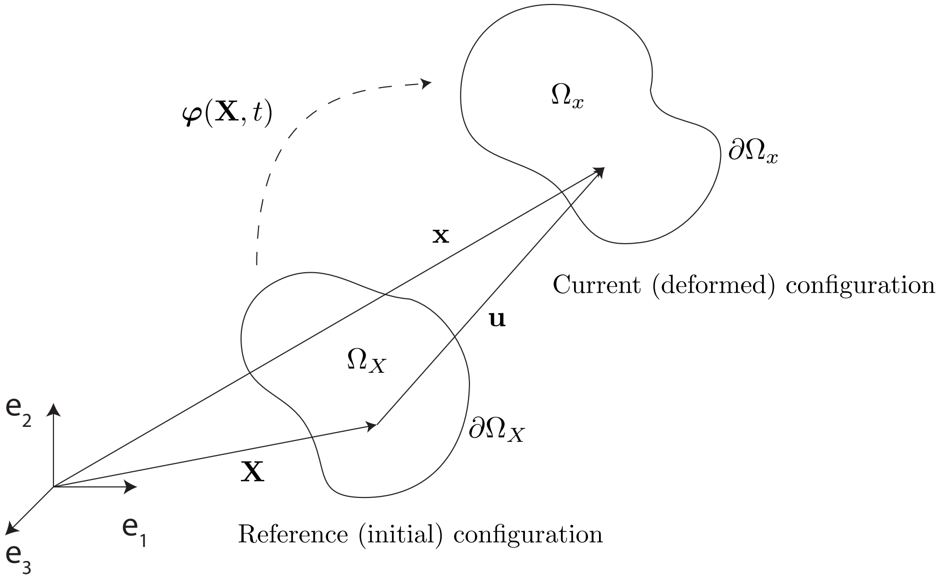


Figure 4.1: Deformation of a material body.

In the total Lagrangian formulation, integrals and derivatives are taken with respect to the material coordinates $\mathbf{X}$ and Lagrangian measures of stress and strain are used [123]. Therefore, considering a body initially occupying a domain Ω_X with undeformed boundary $\partial\Omega_X$ (see Figure 4.1) with initial density, body force and surface traction represented, respectively, by ρ^0, $\mathbf{b}^0$, and

$\mathbf{h}^0$, the equation of motion in the total Lagrangian formulation is [123][55]

$$\int_{\Omega_X} \delta u_i \rho^0 \ddot{u}_i \,\mathrm{d}\Omega + \int_{\Omega_X} \delta F_{ij} P_{ij} \,\mathrm{d}\Omega = \int_{\Omega_X} \delta u_i b_i^0 \,\mathrm{d}\Omega + \int_{\partial\Omega_X} \delta u_i h_i^0 \,\mathrm{d}\partial\Omega, \tag{4.1}$$

where $\mathbf{P}$ is the first Piola-Kirchoff stress tensor and $\mathbf{F}$ is the deformation gradient, defined as $F_{ij} = \partial\varphi_i/\partial X_j$. The updated Lagrangian formulation, instead, is expressed in terms of Eulerian measures of stress and strain, and derivatives and integrals are taken with respect to the spatial coordinates $\mathbf{x}$ [123]. As in the total Lagrangian approach, though, quantities are a function of the material coordinates $\mathbf{X}$. Any generic quantity $\mathbf{f}(\mathbf{X},t)$ can, however, be expressed as a function of spatial coordinates as

$$\mathbf{f}(\mathbf{X},t) = \mathbf{f}(\boldsymbol{\varphi}^{-1}(\mathbf{x},t),t) = \tilde{\mathbf{f}}(\mathbf{x},t) = \tilde{\mathbf{f}}(\boldsymbol{\varphi}(\mathbf{X},t),t), \tag{4.2}$$

where $\mathbf{f} = \tilde{\mathbf{f}} \circ \boldsymbol{\varphi}$ (and $\tilde{\mathbf{f}} = \mathbf{f} \circ \boldsymbol{\varphi}^{-1}$). Alternatively, implicit differentiation can be employed to compute derivatives of functions of $\mathbf{X}$ with respect to $\mathbf{x}$ [123]. In the updated Lagrangian formulation, the equation of motion is expressed as [55][97]

$$\int_{\Omega_x} \delta u_i \rho \ddot{u}_i \,\mathrm{d}\Omega + \int_{\Omega_x} \delta u_{(i,j)} \sigma_{ij} \,\mathrm{d}\Omega = \int_{\Omega_x} \delta u_i b_i \,\mathrm{d}\Omega + \int_{\Gamma_x} \delta u_i h_i \,\mathrm{d}\Gamma, \tag{4.3}$$

where $u_{(i,j)} = (\partial u_j/\partial x_i + \partial u_i/\partial x_j)/2$, $\boldsymbol{\sigma}$ is the Cauchy stress, Ω_x is the deformed domain with current boundary $\partial\Omega_x$ and ρ, $\mathbf{b}$, and $\mathbf{h}$ represent the density, body force, and surface traction in the deformed current configuration, respectively.

The total Lagrangian and the updated Lagrangian descriptions can be shown to be mathematically equivalent through the mapping between the reference and current configurations, and the transformation of stress and strain measures [123]. Choosing one formulation or the other is typically just a matter of convenience based on the employed material constitutive model [97].

4.2 Blending-based reproducing kernel Lagrangian/semi-Lagrangian coupling

The Lagrangian RK approximation approach presented in Section 3.1 can be employed either in a total Lagrangian or updated Lagrangian framework (see Section 4.1), can handle large deformations and is computationally advantageous as it only requires RK neighbour searching and computation of RK shape functions and shape function derivatives with respect to the material coordinates at the beginning of the simulation.

However, for path-dependent material laws such as those that employ, for example, the Cauchy stress as the stress measure, calculation of the spatial derivative of displacement $\mathbf{u}(\mathbf{X},t) \approx \mathbf{u}^h(\mathbf{X},t)$ is required [55][97][77]. If a Lagrangian RK approximation is used, this is obtained through the following chain rule:

$$\begin{aligned}
\frac{\partial u_k^h(\mathbf{X},t)}{\partial x_i} &= \sum_{I=1}^{NP} \frac{\partial \Psi_I^L(\mathbf{X})}{\partial x_i} d_{Ik} = \sum_{I=1}^{NP} \frac{\partial \Psi_I^L(\varphi^{-1}(\mathbf{x}))}{\partial x_i} d_{Ik} \\
&= \sum_{I=1}^{NP} \left[\frac{\partial \Psi_I^L(\varphi^{-1}(\mathbf{x}))}{\partial \varphi_j^{-1}(\mathbf{x})} \frac{\partial \varphi_j^{-1}(\mathbf{x})}{\partial x_i} \right] d_{Ik} \\
&= \sum_{I=1}^{NP} \left[\frac{\partial \Psi_I^L(\mathbf{X})}{\partial X_j} F_{ji}^{-1} \right] d_{Ik},
\end{aligned} \tag{4.4}$$

where $\Psi_I^L(\mathbf{X})$ is the I-th Lagrangian shape function, defined in Chapter 3 (see Section 3.1), and $\mathbf{F}^{-1}$ is typically obtained as the direct inverse of $\mathbf{F}$ [55][97]. Similarly for other fields employed in an updated Lagrangian formulation where, as stated in Section (4.1), the derivatives of functions of $\mathbf{X}$ with respect to $\mathbf{x}$ are computed through implicit differentiation. It is clear from Eq. (4.4) that the Lagrangian RK formulation breaks down when the mapping $\mathbf{x} = \varphi(\mathbf{X},t)$ is no longer one-to-one, hence no longer invertible. This is the case, for example, for problems with changes in topology, such as new surface formation, due to material separation or fragmentation. To circumvent this issue, the Semi-Lagrangian reproducing kernel approximation (see Section

3.2) was developed.

However, this approach breaks down for problems where the mapping φ is no longer one-to-one and the deformation gradient $\mathbf{F}$ is no longer invertible. The semi-Lagrangian RK method (see Section 3.2), on the contrary, since the supports of the kernel functions (and thus of the shape functions) do not deform with the material body and neighbouring points are therefore redefined during the deformation process, is capable of handling such problems However, also the semi-Lagrangian RK method presents some drawbacks. While, on the one hand, it allows, by defining material separation and contact at the discrete level, to model large flows of material motion and to solve contact problems involving arbitrary new surface formation [128][98], it can also lead to undesired numerical fracture. This appears when two RK points nonphysically separate due to insufficient kernel support coverage in the current configuration. Furthermore, the need to perform a search for the neighbouring RK nodes and to reconstruct the semi-Lagrangian shape functions at each integration time step makes the semi-Lagrangian RKPM approach computationally expensive.

In order to retain the advantages of both approaches in a completely meshfree framework, a blending-based Lagrangian/semi-Lagrangian spatial coupling approach is here proposed and investigated.

The main idea behind this method is to use the semi-Lagrangian approximation only in those areas of the domain where the nature of the deformation requires it (e.g. when the deformation gradient $\mathbf{F}$ is no longer invertible) and the Lagrangian RK approximation everywhere else. This is achieved through coupling the semi-Lagrangian and the Lagrangian approximation over a transition region, similarly to the way finite element and element-free Galerkin approximations were coupled in [121]. The approximation over this transition region is obtained by blending the Lagrangian and the semi-Lagrangian RK approximations by means of ramp functions as follows:

$$\begin{aligned}\tilde{f}^h(\varphi(\mathbf{X},t),t) &= [1-r(\mathbf{x})]\tilde{f}^{h,L}(\varphi(\mathbf{X},t),t) + r(\mathbf{x})\tilde{f}^{h,SL}(\mathbf{x},t) \\ &= [1-r(\mathbf{x})]\sum_{I=1}^{NP}\tilde{\Psi}_I^L(\varphi(\mathbf{X},t))\tilde{f}_I(t) \\ &+ r(\mathbf{x})\sum_{I=1}^{NP}\Psi_I^{SL}(\mathbf{x})\tilde{f}_I(t) \\ &= \sum_{I=1}^{NP}\{[1-r(\mathbf{x})]\tilde{\Psi}_I^L(\mathbf{x}) + r(\mathbf{x})\Psi_I^{SL}(\mathbf{x})\}\tilde{f}_I(t) \\ &= \sum_{I=1}^{NP}\{[1-r(\mathbf{x})]\Psi_I^L(\mathbf{X}) + r(\mathbf{x})\Psi_I^{SL}(\mathbf{x})\}\tilde{f}_I(t) \\ &= \sum_{I=1}^{NP}\{[1-r(\mathbf{x})]\Psi_I^L(\varphi^{-1}(\mathbf{x},t)) + r(\mathbf{x})\Psi_I^{SL}(\mathbf{x})\}\tilde{f}_I(t) \\ &= \sum_{I=1}^{NP}\overline{\Psi}_I(\mathbf{x})\tilde{f}_I(t),\end{aligned} \tag{4.5}$$

where $\tilde{\Psi}_I^L(\varphi(\mathbf{X},t))$ are the deformed Lagrangian RK shape functions in the current configuration, so that

$$\tilde{f}^{h,L}(\varphi(\mathbf{X},t),t) = \sum_{I=1}^{NP}\tilde{\Psi}_I^L(\varphi(\mathbf{X},t))\tilde{f}_I(t) = \sum_{I=1}^{NP}\Psi_I^L(\mathbf{X})\tilde{f}_I(t), \tag{4.6}$$

where Eq. (3.22) has been used, and

$$\overline{\Psi}_I(\mathbf{x},t) = [1-r(\mathbf{x})]\Psi_I^L(\varphi^{-1}(\mathbf{x},t)) + r(\mathbf{x})\Psi_I^{SL}(\mathbf{x}) \tag{4.7}$$

with $r(\mathbf{x})$ being a spatial ramp function that is equal to zero for the Lagrangian portion of the domain and equal to unity for the semi-Lagrangian one. In this work a linear ramp function is employed:

$$r(\mathbf{x}) = \begin{cases} 0 & \text{for } w_C \leq d(\mathbf{x}) \\ 1-\frac{d(\mathbf{x})}{w_C} & \text{for } 0 < d(\mathbf{x}) < w_C \\ 1 & \text{for } d(\mathbf{x}) \leq 0 \end{cases}, \tag{4.8}$$

where $w_C \geq 0$ is the constant width of the transition zone, $\partial\Omega^C$ is the boundary of the semi-Lagrangian domain (i.e., the interface between the pure semi-Lagrangian zone and the transition one), and

$$d(\mathbf{x}) = (\mathbf{x}_C - \mathbf{x}) \cdot \mathbf{n}, \tag{4.9}$$

where $\mathbf{n}$ is the outer unit normal to $\partial\Omega^C$, and $\mathbf{x}_C$ is the closest point to $\mathbf{x}$ on $\partial\Omega^C$:

$$\mathbf{x}_C = \{\mathbf{x} \in \partial\Omega^C \mid \min(\| \mathbf{x}_C - \mathbf{x} \|)\}. \tag{4.10}$$

This coupling approach guarantees continuity of the shape functions across the transition zone. This can be seen, for example by looking at Figure 4.3, which shows one-dimensional shape functions for the Lagrangian, transition, and semi-Lagrangian zones in the current configuration $\Omega_x = [0,2]$, obtained after applying a linear displacement $u = X$ to an initial domain $\Omega_X = [0,1]$ (see Figure 4.2 for the shape functions in the undeformed configuration). The interface between the pure semi-Lagrangian zone and the transition one $\partial\Omega^C$ in the current configuration corresponds to $\mathbf{x} = 1.25$ and the width of the transition zone is $w_C = 0.5$. The discretization contains 11 nodes. Linear Lagrangian and semi-Lagrangian are considered, both constructed with cubic B-spline kernel functions with a support size $a = 3.5h$, where h is the nodal spacing in the reference configuration.

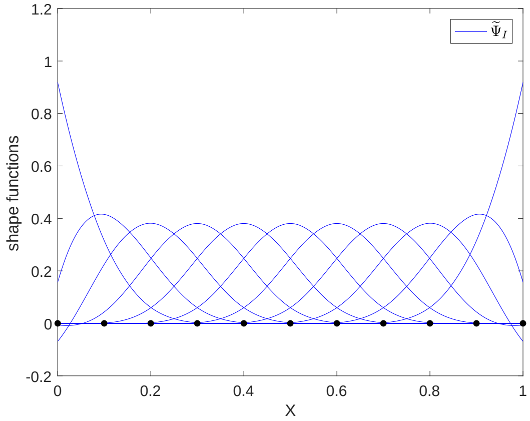


Figure 4.2: Linear RK shape functions in the undeformed configuration.

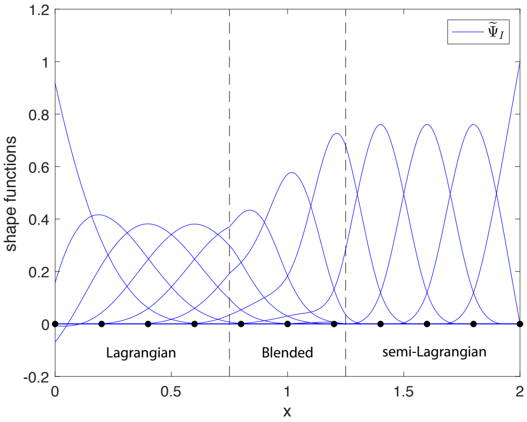


Figure 4.3: Linear Lagrangian, linear semi-Lagrangian, and coupled RK shape functions in the current configuration obtained through application of a linear displacement field.

Continuity of the shape function direct derivatives in the transition zone, however, depends on the continuity of the ramp function. A linear ramp function such as the one in Eq. (4.8) would result in discontinuous derivatives across the transition zone interface [121][127][70]. In this work, therefore, shape function derivatives in the transition zone are computed in the same way as the shape functions, i.e. by blending the the Lagrangian and semi-Lagrangian spatial derivatives:

$$\begin{aligned}\frac{\partial \tilde{f}^h(\varphi(\mathbf{X},t),t)}{\partial x_i} &= [1-r(\mathbf{x})]\frac{\partial \tilde{f}^{h,L}(\varphi(\mathbf{X},t),t)}{\partial x_i} + r(\mathbf{x})\frac{\partial \tilde{f}^{h,SL}(\varphi(\mathbf{X},t),t)}{\partial x_i} \\ &= \sum_{I=1}^{NP}\{[1-r(\mathbf{x})]\frac{\partial \Psi_I^L(\mathbf{X})}{\partial X_j}F_{ji}^{-1} + r(\mathbf{x})\frac{\partial \Psi_I^{SL}(\mathbf{x})}{\partial x_i}\}\tilde{f}_I(t) \\ &= \sum_{I=1}^{NP}\frac{\partial \overline{\Psi}_I(\mathbf{x})}{\partial x_i}\tilde{f}_I(t)\end{aligned} \tag{4.11}$$

,

where

$$\begin{aligned}\frac{\partial \overline{\Psi}_I(\mathbf{x})}{\partial x_i} &= [1-r(\mathbf{x})]\frac{\partial \Psi_I^L(\varphi^{-1}(\mathbf{x},t))}{\partial X_j}F_{ji}^{-1} + r(\mathbf{x})\frac{\partial \Psi_I^{SL}(\mathbf{x})}{\partial x_i} \\ &= [1-r(\mathbf{x})]\frac{\partial \Psi_I^L(\mathbf{X})}{\partial X_j}F_{ji}^{-1} + r(\mathbf{x})\frac{\partial \Psi_I^{SL}(\mathbf{x})}{\partial x_i}.\end{aligned} \tag{4.12}$$

Similarly, in this work, the shape function time derivatives are computed as:

$$\dot{\tilde{f}}^{h}(\varphi(\mathbf{X},t),t) = [1-r(\mathbf{x})]\dot{\tilde{f}}^{h,L}(\varphi(\mathbf{X},t),t) + r(\mathbf{x})\dot{\tilde{f}}^{h,SL}(\varphi(\mathbf{X},t),t), \tag{4.13}$$

where $\dot{\tilde{f}}^{h,L}(\varphi(\mathbf{X},t),t) = \sum_{I=1}^{NP} \Psi_I^L(\mathbf{X})\dot{\tilde{f}}_I(t)$ and $\dot{\tilde{f}}^{h,SL}(\varphi(\mathbf{X},t),t)$ are defined according to Eq. (3.27).

It can be noted that the ramp function $r(\mathbf{x})$ (and hence the domain zones) is defined in the current configuration. This allows for points in the Lagrangian or the transition portion of the domain at a certain time to become part of the semi-Lagrangian zone at a subsequent time. The semi-Lagrangian portion of the domain can be made to increase, for example, by checking in the Lagrangian and mixed portions of the domain values such as the determinant or the condition number of the deformation gradient. If a predefined threshold is reached in a region of the domain, said region can be made part of the semi-Lagrangian zone from then on. It could happen, however, that a portion of the body originally in the semi-Lagrangian zone enters over the course of the deformation into a portion of the domain which is currently defined as Lagrangian. This can happen, for example, in problems involving fragmentation due to high-velocity impacts, where the fragments formed in the semi-Lagrangian zone reach the Lagrangian one. This is avoided by imposing that once a portion of the domain has been semi-Lagrangian, it remains semi-Lagrangian until the end of the simulation, i.e. the semi-Lagrangian portion of the domain is only allowed to grow and not to reduce.

4.3 von Neumann stability analysis of the Lagrangian/semi-Lagrangian coupling approach

In this section, a von Neumann stability analysis is performed to estimate the critical time step of the Lagrangian/semi-Lagrangian coupling approach. The derivation here presented follows from [40] for one dimension. The result of the stability analysis is dependent on the spatial and time integration schemes used. In this work, we consider a stabilized nonconforming nodal integration in space and central difference in time, respectively. Furthermore, for simplicity, a one-dimensional equation of motion is considered.

First, we introduce the Lagrangian/semi-Lagrangian RKPM coupled approximation (see Eqs. (4.5) and (4.11)) in Eq. (4.3) and obtain the semi-discrete equation of motion:

$$\mathbf{M}\ddot{\mathbf{u}} = \mathbf{f}^{ext} - \mathbf{f}^{int}, \tag{4.14}$$

where $\mathbf{f}^{ext}$ is the vector of external forces, $\mathbf{f}^{int}$ is the vector of internal forces where

$$\mathbf{f}_I^{int} = \int_{\Omega_x} \frac{\partial \overline{\Psi}_I(x)}{\partial x} \sigma d\Omega, \tag{4.15}$$

and $\mathbf{M}$ is the mass matrix. We consider a row-sum lumped mass, so that its I-th diagonal value is:

$$\begin{aligned} m_I &= \int_{\Omega_X} \rho^0 \overline{\Psi}_I(X) \sum_{J=1}^{NP} \overline{\Psi}_J(X) d\Omega = \int_{\Omega_X} \rho^0 \overline{\Psi}_I(X) d\Omega \\ &= \int_{\Omega_x} \rho \overline{\Psi}_I(x) d\Omega, \end{aligned} \tag{4.16}$$

where we recall that $\mathbf{x} = \mathbf{X}$ and $\rho = \rho^0$ in the reference configuration. It has to be noted that, since we are considering SNNI integration, the terms associated with the time derivatives of the RK shape functions were omitted in Eq (4.14) (see Section 3.2). The full discrete equation is obtained by applying the central difference scheme to integrate in time:

$$\mathbf{M}\frac{(\mathbf{d}^{n+1} - 2\mathbf{d}^n + \mathbf{d}^{n-1})}{\Delta t^2} = \mathbf{f}^{ext} - \mathbf{f}^{int}, \tag{4.17}$$

where Δt is the time step and n indicates the n-th time step. Due to the diagonal nature of the lumped mass matrix, we can consider a generic I-th row in Eq. (4.17), which leads to

$$m_I \frac{(d_I^{n+1} - 2d_I^n + d_I^{n-1})}{\Delta t^2} = f_I^{ext} - f_I^{int}, \tag{4.18}$$

In order to perform the stability analysis, we introduce a small perturbation in displacement and neglect boundary conditions in Eq. (4.17) [40]:

$$m_I \frac{(\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^n + \tilde{\tilde{d}}_I^{n-1})}{\Delta t^2} = -\tilde{f}_I^{int}, \tag{4.19}$$

where$\tilde{\tilde{}}$denotes perturbation and

$$\tilde{\tilde{f}}_I^{int} = \int_{\Omega_x} \frac{\partial \overline{\Psi}_I(x)}{\partial x} D^\sigma \sum_{J=1}^{NP} \frac{\partial \overline{\Psi}_J(x)}{\partial x} \tilde{\tilde{d}}_J^n d\Omega, \tag{4.20}$$

where D^σ is the material response tensor and

$$\sigma = D^\sigma \sum_{J=1}^{NP} \frac{\partial \overline{\Psi}_J(x)}{\partial x} d_J \tag{4.21}$$

was used in Eq. (4.15). Now, considering a constant D^σ, Eq. (4.19) can be rewritten as:

$$\begin{aligned} \tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^n + \tilde{\tilde{d}}_I^{n-1} &= -\frac{\Delta t^2 D^\sigma}{m_I} \int_{\Omega_x} \frac{\partial \overline{\Psi}_I(x)}{\partial x} \sum_{J=1}^{NP} \frac{\partial \overline{\Psi}_J(x)}{\partial x} \tilde{\tilde{d}}_J^n d\Omega \\ &= -c^2 \Delta t^2 \int_{\Omega_x} \frac{\partial \overline{\Psi}_I(x)}{\partial x} \sum_{J=1}^{NP} \frac{\partial \overline{\Psi}_J(x)}{\partial x} \tilde{\tilde{d}}_J^n d\Omega, \end{aligned} \tag{4.22}$$

where $c^2 = D^\sigma / m_I$. We now consider a uniform discretization with spacing ΔX in the reference configuration, and a uniform deformation with deformation gradient $F = 2$, so that the spacing in the current configuration is $\Delta x = 2\Delta X$. As aforementioned, SNNI is considered for the spatial domain integration. SNNI nonconforming cells are defined in the reference configuration and their size, in this analysis, is kept constant. The SNNI cell size can be defined as being proportional to the corresponding conforming nodal cell [40], which, in the one-dimensional case, corresponds to the nodal spacing ΔX. The proportionality parameter is hereon referred as α (see Figure 4.4).

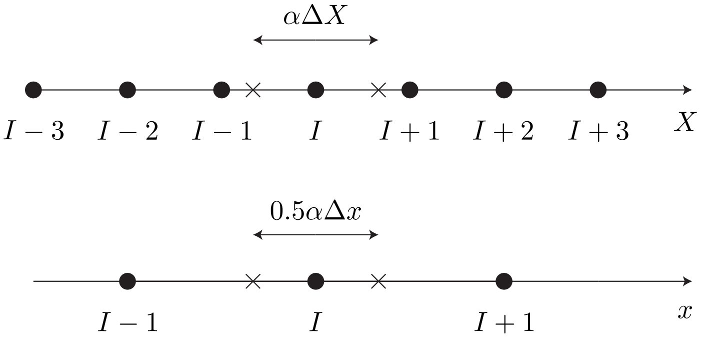


Figure 4.4: SNNI smoothing points associated with node I in the reference and current configuration. The smoothing points are marked by black crosses.

As mentioned in Section 3.2, in the SNNI approach direct shape function gradients are replaced by smoothed gradients, obtained by smoothing over nonconforming nodal cells (see Figures 3.4 and 4.4). Given that the coupled gradient is computed by coupling of the Lagrangian and semi-Lagrangian gradients, the smoothed Lagrangian and semi-Lagrangian gradients are first computed and then combined according to Eq. 4.12. For the one-dimensional case with uniform distribution, the boundary of the smoothing domain for a generic node L is $[X_L^-, X_L^+]$ in the reference configuration and $[x_L^-, x_L^+]$ in the current configuration, where $X_L^- = X_L - 0.5\alpha\Delta X$, $X_L^+ = X_L + 0.5\alpha\Delta X$, $x_L^- = x_L - 0.5\alpha\Delta X = x_L - 0.25\alpha\Delta x$, and $x_L^+ = x_L + 0.5\alpha\Delta X = x_L + 0.25\alpha\Delta x$. The Lagrangian smoothed gradient and the semi-Lagrangian smoothed gradient are, respectively

$$\frac{\partial \overline{\overline{\Psi}}_I^L(X_L)}{\partial X} = \frac{\Psi_I^L(X_L^+) - \Psi_I^L(X_L^-)}{\alpha\Delta X}, \tag{4.23}$$

and

$$\frac{\partial \overline{\overline{\Psi}}_I^{SL}(x_L)}{\partial x} = \frac{\Psi_I^{SL}(x_L^+) - \Psi_I^{SL}(x_L^-)}{\alpha\Delta x}, \tag{4.24}$$

making the smoothed coupled gradient

$$\begin{aligned}\frac{\partial\overline{\overline{\Psi}}_I(x_L)}{\partial x} &= [1-r]\left[\frac{\Psi_I^L(X_L^+)-\Psi_I^L(X_L^-)}{\alpha\Delta X}\right]F^{-1} \\ &+ r\left[\frac{\Psi_I^{SL}(x_L^+)-\Psi_I^{SL}(x_L^-)}{\alpha\Delta x}\right],\end{aligned} \tag{4.25}$$

where, following [40], $r(x)$ has been taken to be constant. It has to be noted that

$$\begin{aligned}\frac{\Psi_I^L(X_L^+)-\Psi_I^L(X_L^-)}{\alpha\Delta X}F^{-1} &= \frac{\Psi_I^L(X_L^+)-\Psi_I^L(X_L^-)}{\alpha\Delta x} \\ &= \frac{\tilde{\Psi}_I^L(\varphi(X_L^+,t))-\tilde{\Psi}_I^L(\varphi(X_L^-,t))}{\alpha\Delta x}.\end{aligned} \tag{4.26}$$

In the first term of Eq (4.26), the smoothed gradient of the Lagrangian shape functions is evaluated in the undeformed configuration and pushed forward through the inverse of the deformation gradient. The last term represents the smoothed gradient of the deformed Lagrangian shape function computed over a deformed smoothing cell, which is obtained by applying the deformation gradient to the SNNI smoothing cell in the reference configuration. These two approaches are therefore equivalent. This results in the use of two different smoothing cells: a deformed one for the Lagrangian smoothed gradient contribution and an undeformed one for the semi-Lagrangian smoothed gradient contribution (see Figure 4.5).

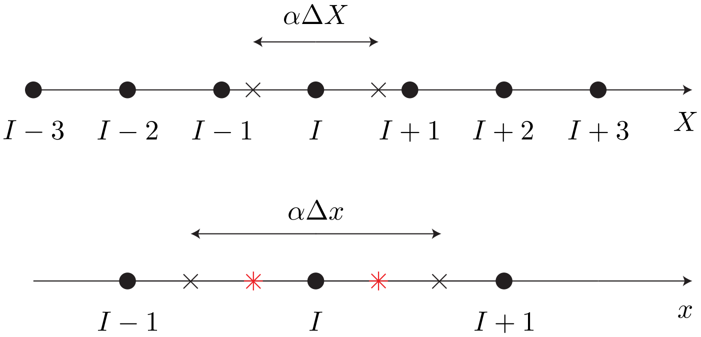


Figure 4.5: SNNI smoothing points associated with node I for the Lagrangian (indicated as black crosses) and semi-Lagrangian (marked by red asterisks) contributions to the coupled smoothed gradient.

Now, by applying the stabilized nonconforming nodal integration to the right-hand side of Eq. (4.22), we obtain

$$\begin{aligned}\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^{n} + \tilde{\tilde{d}}_I^{n-1} = -c^2\Delta t^2 \sum_{J=1}^{NP}\sum_{L=1}^{NP} & \left\{ [1-r]\left[\frac{\Psi_I^L(X_L^+) - \Psi_I^L(X_L^-)}{\alpha\Delta X}\right]F^{-1} \right. \\ & \left. + r\left[\frac{\Psi_I^{SL}(x_L^+) - \Psi_I^{SL}(x_L^-)}{\alpha\Delta x}\right]\right\} \\ & \cdot\left\{ [1-r]\left[\frac{\Psi_J^L(X_L^+) - \Psi_J^L(X_L^-)}{\alpha\Delta X}\right]F^{-1} \right. \\ & \left. + r\left[\frac{\Psi_J^{SL}(x_L^+) - \Psi_J^{SL}(x_L^-)}{\alpha\Delta x}\right]\right\}\Delta x \tilde{\tilde{d}}_J^{n},\end{aligned} \tag{4.27}$$

$$\begin{aligned}\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^{n} + \tilde{\tilde{d}}_I^{n-1} = -c^2\Delta t^2 \sum_{J=1}^{NP}\sum_{L=1}^{NP} & \left\{ [1-r]\left[\frac{\Psi_I^L(X_L^+) - \Psi_I^L(X_L^-)}{\alpha\Delta x}\right] \right. \\ & \left. + r\left[\frac{\Psi_I^{SL}(x_L^+) - \Psi_I^{SL}(x_L^-)}{\alpha\Delta x}\right]\right\} \\ & \cdot\left\{ [1-r]\left[\frac{\Psi_J^L(X_L^+) - \Psi_J^L(X_L^-)}{\alpha\Delta x}\right] \right. \\ & \left. + r\left[\frac{\Psi_J^{SL}(x_L^+) - \Psi_J^{SL}(x_L^-)}{\alpha\Delta x}\right]\right\}\Delta x \tilde{\tilde{d}}_J^{n},\end{aligned} \tag{4.28}$$

$$\begin{aligned}\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^{n} + \tilde{\tilde{d}}_I^{n-1} = -\frac{c^2\Delta t^2}{\alpha^2\Delta x} \sum_{J=1}^{NP}\sum_{L=1}^{NP} & \{[1-r][\Psi_I^L(X_L^+) - \Psi_I^L(X_L^-)] \\ & + r[\Psi_I^{SL}(x_L^+) - \Psi_I^{SL}(x_L^-)]\} \\ & \cdot\{[1-r][\Psi_J^L(X_L^+) - \Psi_J^L(X_L^-)] \\ & + r[\Psi_J^{SL}(x_L^+) - \Psi_J^{SL}(x_L^-)]\}\tilde{\tilde{d}}_J^{n}.\end{aligned} \tag{4.29}$$

Considering RKPM shape functions with support size $a \leq (2+0.5\alpha)\Delta X = (1+0.25\alpha)\Delta x$, $\alpha \in [1,2]$, and by letting the perturbed displacement take the plane wave form

$$\tilde{\tilde{d}}_I^{n} = \lambda^n e^{ik(I\Delta x)}, \tag{4.30}$$

where k is the wave-number and λ is the amplification factor, the following stability criterion can be obtained

$$c^2\Delta t^2 \leq \frac{\alpha^2\Delta x}{\bar{A}}, \tag{4.31}$$

where

$$\bar{A} = \left[(1-r)A^L + rA^{SL}\right]^2, \tag{4.32}$$

with

$$\begin{aligned} A^L = [\Psi^L((3-0.5\alpha)\Delta X)\sin(3\bar{k}\Delta X) + \Psi^L((2-0.5\alpha)\Delta X)\sin(2\bar{k}\Delta X) \\ + \Psi^L((1-0.5\alpha)\Delta X)\sin(\bar{k}\Delta X) - \Psi^L((1+0.5\alpha)\Delta X)\sin(\bar{k}\Delta X), \end{aligned} \tag{4.33}$$

with $\bar{k} = 2k$, and

$$A^{SL} = [\Psi^{SL}((1-0.25\alpha)\Delta x)\sin(k\Delta x)]. \tag{4.34}$$

The details of the derivation are reported in 4.5. It has to be noted that in the stability criterion (see Eq. (4.31)), $\bar{A}$ would be replaced by ${A^L}^2$ and ${A^{SL}}^2$ in the case of a purely Lagrangian or semi-Lagrangian approximation, respectively. In 4.5 it is shown that

$$\bar{A} = \left[(1-r)A^L + rA^{SL}\right]^2 \leq \max({A^L}^2, {A^{SL}}^2), \tag{4.35}$$

meaning that the critical time step for the coupled case is larger than the larger critical time step between the purely Lagrangian and the purely semi-Lagrangian cases. This suggests that, for a given problem, selecting a time step for which the purely Lagrangian and semi-Lagrangian cases are stable, would guarantee stability of the coupled case as well.

4.4 Numerical results

In this section, the effectiveness of the proposed Lagrangian/semi-Lagrangian RKPM coupling is verified through a set of numerical examples. Its efficiency compared to fully semi-Lagrangian RK simulations is also demonstrated through run-time comparisons. All results in this section are obtained for an MPI implementation of the 3D RKPM formulation in FORTRAN, run on a single compute node (RAM 256 GB) containing 32 processors (2.3GHz). The amount of processors used is specified in each example. For time integration, central difference with lumped mass is employed.

4.4.1 Wave propagation in an elastic bar

Consider the following one-dimensional undamped wave propagation problem in an elastic bar with $\Omega_X = [0,1]$ and time $t \in (0,T)$:

$$\ddot{u}(X,t) = c^2 \frac{d^2u(X,t)}{dX^2} \quad \text{for } X \in \Omega_X,\ t \in (0,T)\ , \tag{4.36}$$

with boundary conditions

$$\begin{cases} u(0,t) = 0 \\ u(1,t) = u_0 \sin(\frac{\pi c t}{2}) \end{cases} , \tag{4.37}$$

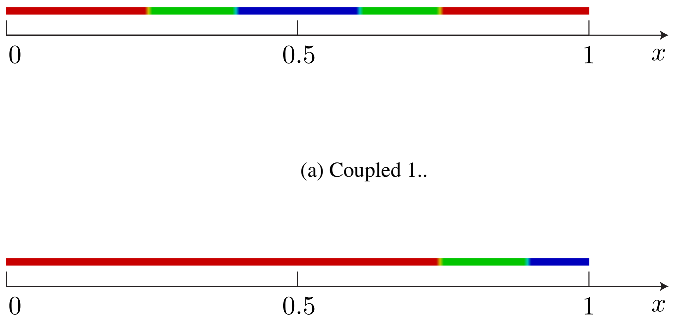
and initial conditions

$$\begin{cases} u(X,0) = 0 \\ \dot{u}(X,0) = u_0(\frac{\pi c}{2}) \sin(\frac{\pi X}{2}) \end{cases} , \tag{4.38}$$

where $u_0 = 0.001\ m$, $c = \sqrt{\frac{E}{\rho}}$, E is the Young's modulus and ρ is the density of the bar. The exact solution for this problem is

$$u(X,t) = u_0 \sin(\frac{\pi X}{2}) \sin(\frac{\pi c t}{2}) \tag{4.39}$$

We built a model of a bar of length 1 m with a cross section 0.0125 m by 0.0125 m. Poisson's ration $\nu = 0$ was used to simulate a one-dimensional solution. Young's modulus and density of the bar are $E = 78.2\ Pa$ and $\rho = 2700\ \frac{kg}{m^3}$, respectively. The employed discretization consists of a total of 324 RK nodes (four nodes in the cross sectional area). We consider the following approximations: full semi-Lagrangian RK, full Lagrangian RK, and two coupled cases (hereon referred as “Coupled 1” and “Coupled 2”). For all cases we employ SNNI, linear basis and a cubic B-spline kernel function with a brick support. The kernel support in the i-th direction is taken to be $a_i = 2.5h_i$ where h_i is the nodal spacing in the i-th direction, with $i = 1,2,...,3$. In the first coupled case "Coupled 1" the semi-Lagrangian portion of the domain is positioned centered around the bar midpoint. In this case the semi-Lagrangian zone $\Omega^{SL} = [0.4, 0.6]$ for all $t \in (0, T)$, meaning that $\partial\Omega^C = \{x = 0.4\} \cup \{x = 0.6\}$. The width of the transition zone is $w_c = 0.15$. In "Coupled 2", the semi-Lagrangian zone is at the free-end of the bar: $\Omega^{SL} = [0.9, 1]$ ($\partial\Omega^C = \{x = 0.9\}$), and the width of the transition zone is still $w_c = 0.15$. Figure 4.6 shows the Lagrangian, semi-Lagrangian and transition zones.



(a) Coupled 1..

(b) Coupled 2.

Figure 4.6: Lagrangian (red), semi-Lagrangian (blue) and transition (green) zones.

Figure 4.7 and Figure 4.8 show the displacement time histories of the bar end ($X = 1$) and of the bar midpoint ($X = 0.5$), respectively. The employed time step Δt is equal to 10^{-6} s. Such time step was chosen so that the purely Lagrangian and semi-Lagrangian simulations would be stable. Consistently with the analysis in Section 4.3, this choice resulted in a stable coupled simulation. It can be observed that all the numerical approximations agree well with the exact solution.

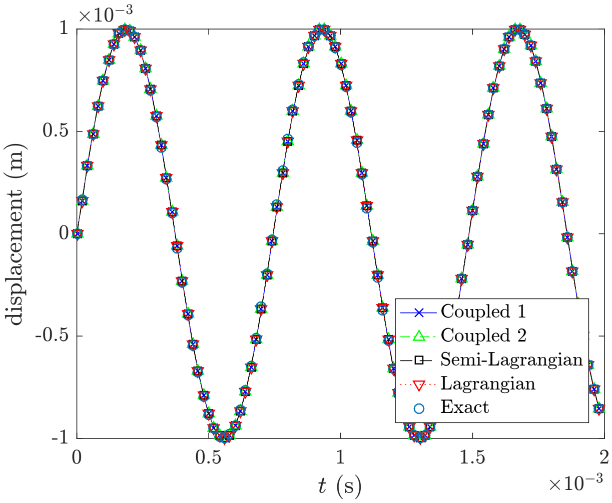


Figure 4.7: Displacement time history for the bar tip end ($X = 1$).

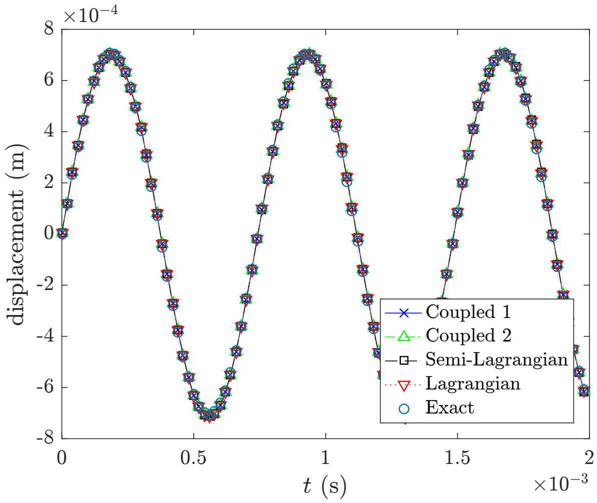


Figure 4.8: Displacement time history for the bar midpoint ($X = 0.5$).

Table 4.1 shows the run-time for the various considered approximations for two runs, each performed on four processors. As expected the purely Lagrangian approximation simulation is the fastest one. In “Coupled 1”, where the percentage of domain where the semi-Lagrangian shape functions had to be recomputed (i.e. the semi-Lagrangian and transition zones) is equal to 50% of the total domain, the total run-time is on average equal to 63.96% of the run-time of the full semi-Lagrangian RK simulation. In “Coupled 2”, instead, the semi-Lagrangian and transition zones amount to 25% of the total domain and the total run-time is on average equal to 44.43% of the run-time of the full semi-Lagrangian RK simulation.

Table 4.1: Run-times for the one-dimensional wave propagation problem.

-	Run-time (s)	
Method	Run 1	Run 2
Lagrangian	6.286	6.192
semi-Lagrangian	25.709	26.342
Coupled 1	16.412	16.882
Coupled 2	11.590	11.536

4.4.2 Taylor bar impact

In this section, we consider a cylindrical aluminium bar impacting a rigid frictionless wall [34]. Experimental results for this problem are provided in [86]. The initial height and radius of the bar are 2.346 cm and 0.391 cm, respectively. We define a reference system $[x,y,z]$ where x is along the longitudinal direction of the bar. The bottom and top side of the undeformed bar are at $x = 0$ and $x = 2.346$, respectively. The initial velocity of the bar is $\mathbf{v_0} = (-373,0,0)$ m/s. The Young’s modulus, density and Poisson’s ratio are $E = 1$ GPa, $\nu = 0.3$, and $\rho = 2700$ kg/m^3, respectively.

We consider a J_2 plasticity constitutive model with isotropic hardening. The yield function is [56][75]

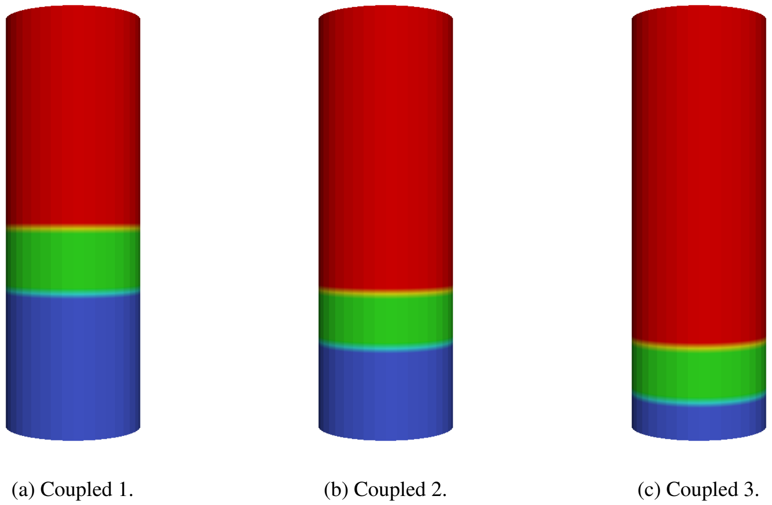
$$f(\mathbf{s},\bar{e}^p) = \|\mathbf{s}\| - \sqrt{\frac{2}{3}}K(\bar{e}^p), \tag{4.40}$$

where $\mathbf{s}$ is the deviatoric part of the Cauchy stress, $\bar{e}^p$ is the equivalent plastic strain and

$$K(\bar{e}^p) = \sigma_Y(1+125\bar{e}^p)^{0.1} \tag{4.41}$$

We consider an initial yield stress $\sigma_Y = 0.29$ GPa and a plastic hardening modulus $H = 0$ *GPa* [56]. The impact of the aluminium bar against the wall is modelled by imposing a zero displacement along the x direction to the nodes at $x = 0$ in the undeformed configuration. The bar is modelled using 185 nodes in the cross-section and 41 nodes in the longitudinal direction (for a total of 7,585 nodes).

First, we consider the following cases: full Lagrangian RK, full semi-Lagrangian RK, and three coupled cases (hereon referred as "Coupled 1", "Coupled 2" and "Coupled 3"). For all cases we employ SNNI, linear basis and a cubic B-spline kernel function with a brick support. The kernel support in the i-th direction is taken to be $a_i = 2.5h_i$ where h_i is the nodal spacing in the i-th direction, with $i = 1,2,...,3$. In all the coupled cases, the semi-Lagrangian portion of the domain is positioned so that it contains the bottom side of the bar. In "Coupled 1" the semi-Lagrangian zone Ω^{SL} is defined to be between the yz planes at $x = 0$ and $x = 0.87$ for all $t \in (0,T)$, while in "Coupled 2" the semi-Lagrangian zone Ω^{SL} is defined to be between the yz planes at $x = 0$ and $x = 0.57$ for all $t \in (0,T)$. In "Coupled 3", Ω^{SL} is between the yz planes at $x = 0$ and $x = 0.27$ for all $t \in (0,T)$. The width of the transition zone is in all cases $w_c = 0.3$. Figure 4.9 shows the coupling zones in the initial undeformed configuration, while Figure 4.10 shows the deformed Taylor bar with the Lagrangian, semi-Lagrangian and transition zones. Table 4.2 contains the final deformed bar heights and radii for the considered cases, as well as the experimentally observed final height reported in [86]. All cases get reasonably close to the experimental value.

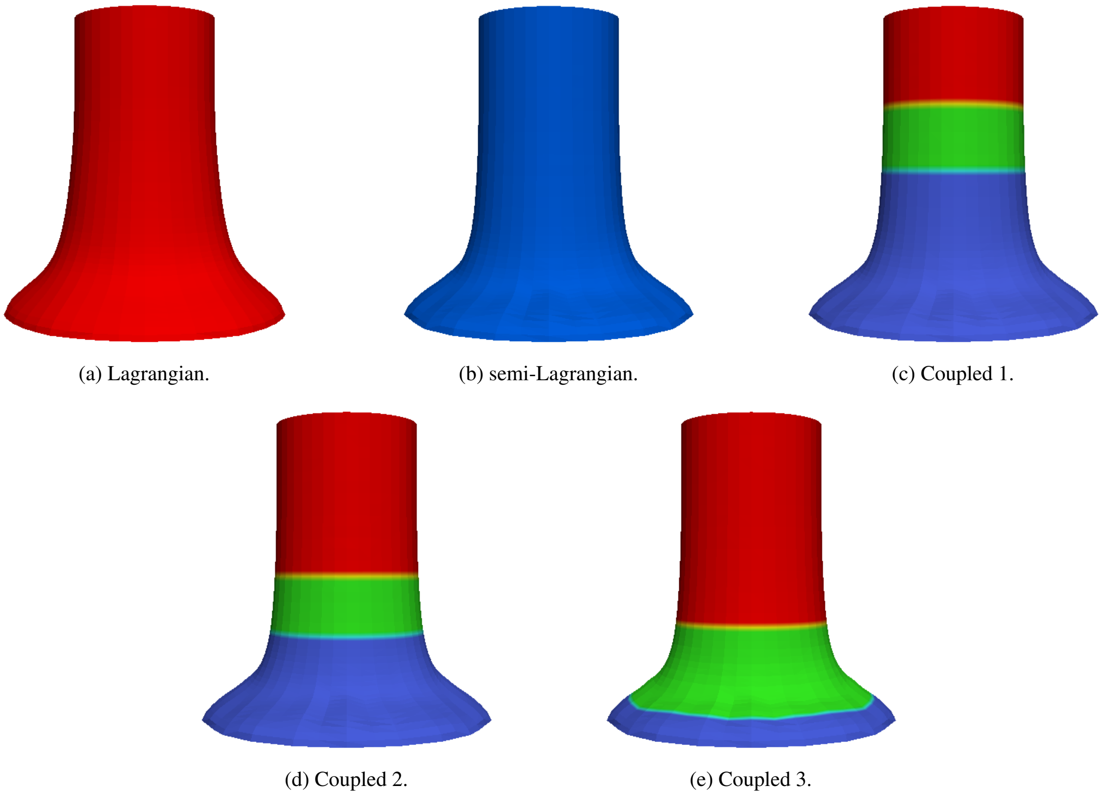


(a) Coupled 1.

(b) Coupled 2.

(c) Coupled 3.

Figure 4.9: Initial Taylor bar configurations in the coupling approach. The semi-Lagrangian, Lagrangian and transition zones are represented in blue, red, and green, respectively.



(a) Lagrangian.

(b) semi-Lagrangian.

(c) Coupled 1.

(d) Coupled 2.

(e) Coupled 3.

Figure 4.10: Deformed Taylor bars. The semi-Lagrangian, Lagrangian and transition zones are represented in blue, red, and green, respectively.

Table 4.2: Dimensions of the deformed Taylor Bar.

Method	Radius (cm)	Height (cm)
Lagrangian	0.769	1.632
semi-Lagrangian	0.774	1.630
Coupled 1	0.774	1.630
Coupled 2	0.774	1.630
Coupled 3	0.772	1.630
Experiment [86]	-	1.651

It has to be noted that the coupled cases provide results similar to the full semi-Lagrangian formulation. This is due to the fact that, in the coupled approaches, the semi-Lagrangian zone was placed where the majority of the deformation happens (i.e. at the bottom of the bar). However, it can be noted in Table 4.3 that the "Coupled 1", "Coupled 2", and "Coupled 3" cases had a run-time equal to 80.8%. 71.1%, 61.6%, respectively, compared to the run-time of the full semi-Lagrangian simulation. It has to be noted, also, that the size of the definition of the Lagrangian, semi-Lagrangian and transition zones, was kept constant for the whole duration of the simulations (see Figures 4.10 and 4.9). For this reason, as the deformation occurs, portions of the domain originally in the Lagrangian zone entered the transition and semi-Lagrangian ones, requiring shape functions re-computation. The runs were performed with 16 processors for a simulation end-time $T = 3.15 \cdot 10^{-5}\ s$ and a time step $\Delta t = 10^{-9}\ s$. This Δt was selected as it resulted in stable purely Lagrangian and semi-Lagrangian simulations. Consistently with the analysis in Section 4.3, this also resulted in a stable coupled simulation.

Table 4.3: Run-times for the Taylor bar impact problem.

Method	Run-time (s)
Lagrangian	2173.0
semi-Lagrangian	14772.2
Coupled 1	11936.4
Coupled 2	10506.2
Coupled 3	9104.7

4.4.3 Impact-perforation simulation

In this section, the penetration of a spherical steel projectile impacting a CorTuf [21] ultra high-strength concrete panel with a zero degree angle is considered. This problem is taken from the experiments presented in [63], and the corresponding experimental results are used as a reference for comparison with the the numerical results from the RKPM simulation. Specifically, impact of a spherical projectile with diameter equal to 1.11 cm and an initial velocity of 1114.65 m/s on a 2.54 cm thick CorTuf concrete panel is considered. The width and the height of the panel are both equal to 30.48 cm.

In the simulation, the projectile is modeled using 1163 nodes while the concrete panel was discretized using 190000 nodes. The projectile is modeled using J2 plasticity with properties reported in Table 4.4, while concrete is modeled using the MIDM (microcrack informed damage model)-enhanced AFC model. The AFC material model parameters can be found in [63].

Table 4.4: Spherical projectile properties.

Property	Value
Young's modulus, E	200 GPa
Poisson's ratio, ν	0.26
Density, ρ	7806 kg/m^3
Mass proportional damping	0.0001 1/s
Initial yield stress σ_Y	2400 MPa
Hardening modulus H	2500 MPa

Contact between the panel and the projectile was modeled using the kernel contact approach employed in [98]. Quasi-linear approximation functions are employed with linear basis, $\alpha = 0.0001$ and $\varepsilon = 0.005$. Quartic B-spline kernel functions are used. It was noted in [63] that, in order to prevent nonphysical fragmentation of the projectile, a sufficiently large support size has to be considered. In this work, we take the normalized support size to be equal to 3.0 for the projectile and 1.5 for the concrete panel.

Two cases are considered: a full semi-Lagrangian simulation and a coupled case. For the coupled case a right circular cylinder defines the semi-Lagrangian zone Ω^{SL}. The axis of the cylinder is positioned so that it connects the centres of the impact and exit faces of the concrete panel. The radius of the base of the cylinder is taken to be equal to 5 cm, while the width of the transition zone is $w_c = 2$ cm. The Lagrangian, semi-Lagrangian and transition zones are shown in Figure 4.11.

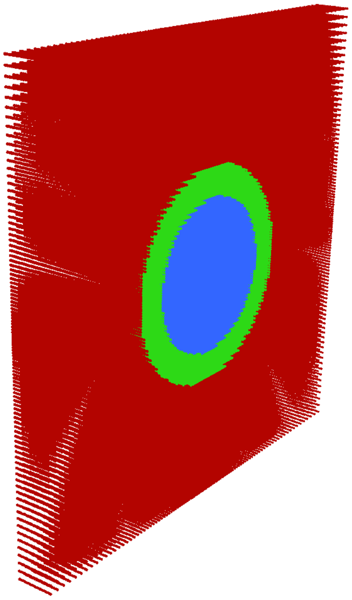
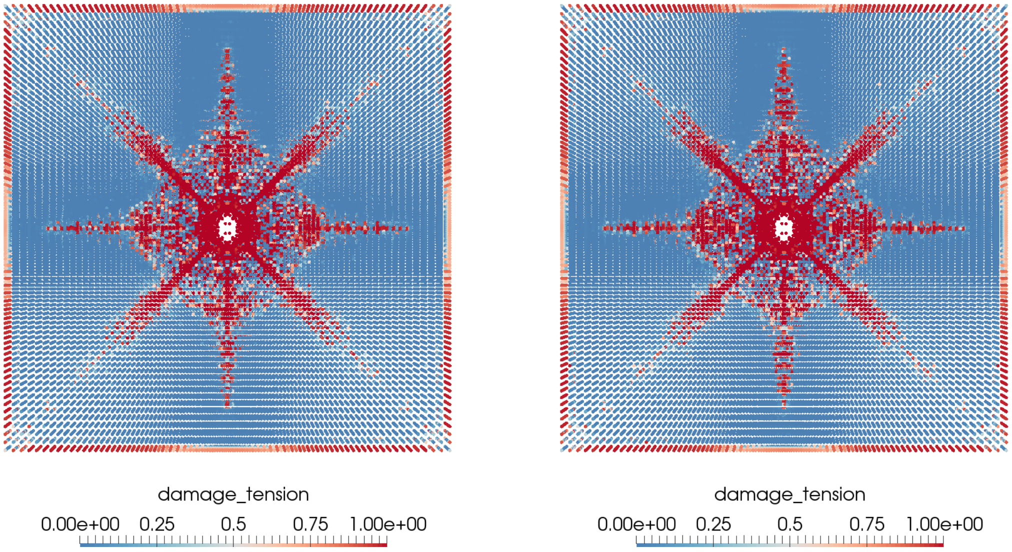


Figure 4.11: Semi-Lagrangian (blue), transition (green) and Lagrangian (red) RK zones.

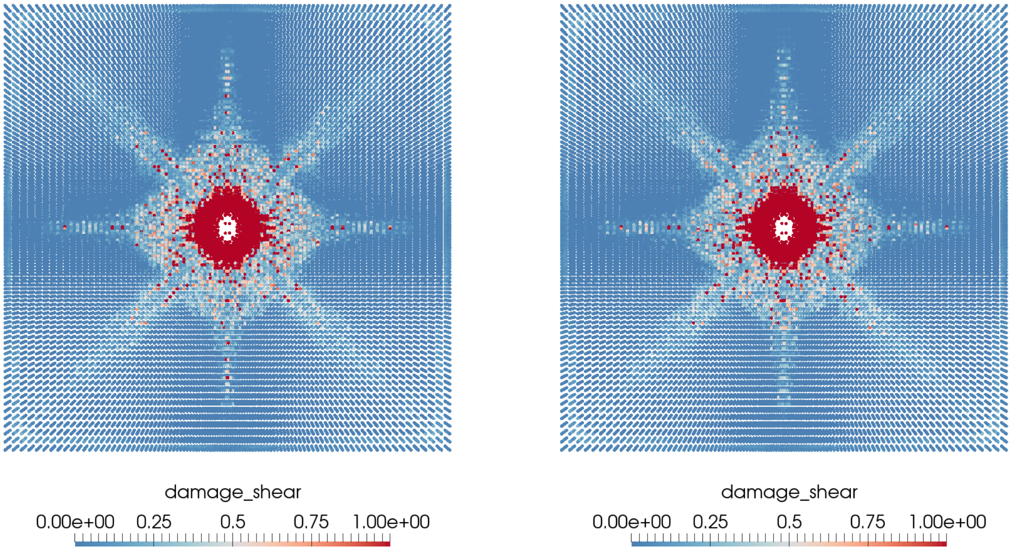
The tensile damage pattern and the shear damage pattern at the panel exit face are shown in Figures 4.12 and 4.13, respectively, while the corresponding debris clouds are shown in Figure 4.14. The exit velocities of the projectile are reported in Table 4.5. It can be observed that the results obtained with the semi-Lagrangian RK simulation and with the coupled one are similar. However, as shown in Table 4.6, the "Coupled" case took 63.8% of the run-time of the full semi-Lagrangian simulation. The runs were performed with 32 processors for a simulation end-time $T = 39 \cdot 10^{-5}$ s and a time step $\Delta t = 10^{-8}$ s. Such time step was chosen so that the purely Lagrangian and semi-Lagrangian simulations would be stable. Consistently with the analysis in Section 4.3, this choice resulted in a stable coupled simulation.



(a) Coupled.

(b) semi-Lagrangian.

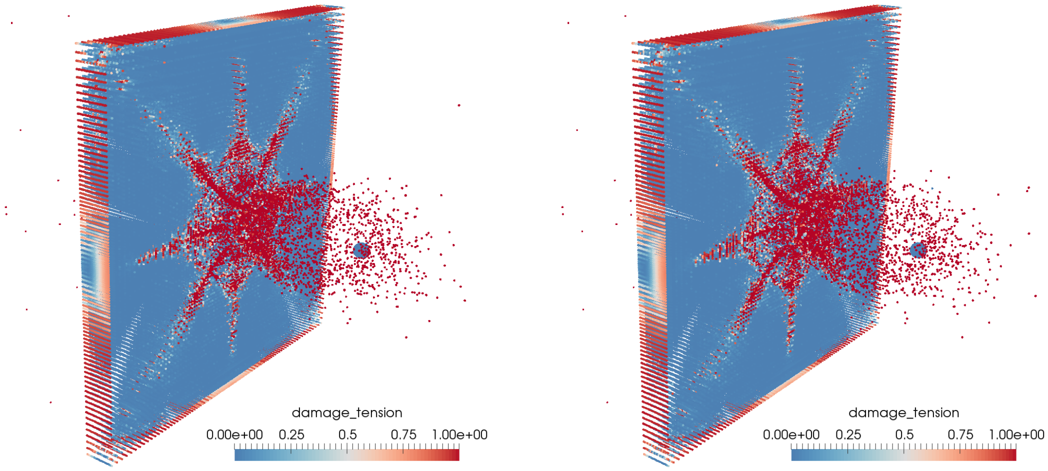
Figure 4.12: Tensile damage at panel the exit face.



(a) Coupled.

(b) semi-Lagrangian.

Figure 4.13: Shear damage at panel the exit face.


damage_tension
0.00e+00 0.25 0.5 0.75 1.00e+00
damage_tension
0.00e+00 0.25 0.5 0.75 1.00e+00

(a) Coupled.

(b) semi-Lagrangian.

Figure 4.14: Debris clouds due to concrete perforation at $t = 39 \cdot 10^{-5}$ s.

Table 4.5: Projectile exit velocity.

Method	Exit Velocity (m/s)	Velocity reduction (%)
semi-Lagrangian	338.0	69.7
Coupled	338.3	69.6
Experiment [63]	544.7	51.13

Table 4.6: Run-times for the impact-perforation problem.

Method	Run-time (s)
semi-Lagrangian	84970.8
Coupled	54197.5

Acknowledgments

This chapter is currently being prepared for submission for publication of the material. M. Pasetto, J. Baek, J.S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, “A Lagrangian/semi-Lagrangian Coupling Approach for Meshfree Modeling of Extremely Large Deformation Problems”. The dissertation author was the primary investigator of this material.

4.5 Appendix 4-A: von Neumann stability analysis expansions

In order to obtain the stability criterion for the Lagrangian/semi-Lagrangian coupled approximation, we start by expanding the right-hand side of Eq. (4.29). Since the considered discretization is uniform and we ignore any influence coming from the boundary of the domain, we can exploit the symmetrical nature of the RKPM shape functions, as well as the fact that the all shape functions have the same values for the same distances (e.g., $\Psi_I(p\Delta X) = \Psi_J(p\Delta X) = \Psi(p\Delta X)$, $\forall I, J$ with $p \in \mathbb{R}$). Recalling that the considered support size for the RK shape functions is taken to be $a \leq (2+0.5\alpha)\Delta X = (1+0.25\alpha)\Delta x$ with $\alpha \in [1,2]$, and by defining $r_1 = 1-r$ and $r_2 = r$, we obtain

$$
\begin{aligned}
&[\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^{n} + \tilde{\tilde{d}}_I^{n-1}]\frac{\alpha^2\Delta x}{-c^2\Delta t^2} = \qquad\qquad (B.1)\\
&\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-6}^{n}\\
&+\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-5}^{n}\\
&+\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-5}^{n}\\
&+\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\\
&\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I-4}^{n}\\
&+\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-4}^{n}\\
&+\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot\\
&\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-4}^{n}\\
&+\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\\
&\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I-3}^{n}\\
&+\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot\\
&\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-3}^{n}\\
&+\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\\
&\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[+\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I-2}^{n}\\
&\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[+\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot\\
&\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I-2}^{n}\\
&+\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot\\
&\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-2}^{n}\\
&+\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I-1}^{n}\\
&+\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot
\end{aligned}
$$

$$\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[+\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}^n_{I-1}$$

$$+\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I-1}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I-1}$$

$$+\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[\Psi^L((1-0.5\alpha)\Delta X)+\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[\Psi^L((1+0.5\alpha)\Delta X)+\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}^2\tilde{\tilde{d}}^n_{I}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I+1}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot$$

$$\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}^n_{I+1}$$

$$+\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[+\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I+1}$$

$$+\{r_1[+\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I+1}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot$$

$$\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}^n_{I+2}$$

$$\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}^n_{I+2}$$

$$+\{r_1[-\Psi^L((1+0.5\alpha)\Delta X)+\Psi^L((1-0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}^n_{I+2}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot$$

$$\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I+3}^{n}$$

$$+\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+3}^{n}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot$$

$$\{r_1[\Psi^L((1-0.5\alpha)\Delta X)-\Psi^L((1+0.5\alpha)\Delta X)]+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\tilde{\tilde{d}}_{I+4}^{n}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+4}^{n}$$

$$+\{r_1[\Psi^L((1+0.5\alpha)\Delta X)-\Psi^L((1-0.5\alpha)\Delta X)]+r_2[-\Psi^{SL}((1-0.25\alpha)\Delta x)]\}\cdot$$

$$\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+4}^{n}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((2-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+5}^{n}$$

$$+\{r_1[-\Psi^L((2-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+5}^{n}$$

$$+\{r_1[-\Psi^L((3-0.5\alpha)\Delta X)]\}\cdot\{r_1[\Psi^L((3-0.5\alpha)\Delta X)]\}\tilde{\tilde{d}}_{I+6}^{n}$$

$$
\begin{aligned}
&\tilde{\tilde{d}}_I^{n+1} - 2\tilde{\tilde{d}}_I^{n} + \tilde{\tilde{d}}_I^{n-1} = \\
&-\frac{c^2 \Delta t^2}{\alpha^2 \Delta x}\{[-r_1^2 {\Psi^{L}}^{2}((3-0.5\alpha)\Delta X)]\tilde{\tilde{d}}_{I-6}^{n} \\
&+[-2r_1^2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X)]\tilde{\tilde{d}}_{I-5}^{n} \\
&+[2r_1^2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1^2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&-2r_1 r_2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&-r_1^2 {\Psi^{L}}^{2}((2-0.5\alpha)\Delta X)]\tilde{\tilde{d}}_{I-4}^{n} \\
&+[2r_1^2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1^2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&-2r_1 r_2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I-3}^{n} \\
&+[-2r_1^2 \Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{L}((3-0.5\alpha)\Delta X) \\
&+2r_1^2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&+2r_1 r_2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&+r_1^2 \Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1 r_2 \Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&+2r_1 r_2 \Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&-r_1^2 {\Psi^{L}}^{2}((1-0.5\alpha)\Delta X) - r_1^2 {\Psi^{L}}^{2}((1+0.5\alpha)\Delta X) \\
&-r_2^2 {\Psi^{SL}}^{2}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I-2}^{n} \\
&+[2r_1^2 \Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X) \\
&+2r_1^2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&-2r_1^2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&+2r_2 r_2 \Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I-1}^{n}
\end{aligned}
\tag{B.2}
$$

$$
\begin{aligned}
&+[2r_1^2\Psi^{L^2}((3-0.5\alpha)\Delta X)+2r_1^2\Psi^{L^2}((2-0.5\alpha)\Delta X) \\
&+2r_1^2\Psi^{L^2}((1-0.5\alpha)\Delta X)+2r_1^2\Psi^{L^2}((1+0.5\alpha)\Delta X) \\
&+2r_2^2\Psi^{SL^2}((1-0.25\alpha)\Delta x) \\
&-4r_1^2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&+4r_1r_2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&-4r_1r_2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_I^n \\
&+[2r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X) \\
&+2r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&-2r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&+2r_2r_2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I+1}^n \\
&+[-2r_1^2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{L}((3-0.5\alpha)\Delta X) \\
&+2r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&+2r_1r_2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&+r_1^2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1r_2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&+2r_1r_2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x) \\
&-r_1^2\Psi^{L^2}((1-0.5\alpha)\Delta X)-r_1^2\Psi^{L^2}((1+0.5\alpha)\Delta X) \\
&-r_2^2\Psi^{SL^2}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I+2}^n \\
&+[2r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X) \\
&-2r_1r_2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)]\tilde{\tilde{d}}_{I+3}^n \\
&+[2r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X) \\
&-2r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)
\end{aligned}
$$

$$
\begin{aligned}
&-2r_1r_2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)\\
&-r_1^2\Psi^{L^2}((2-0.5\alpha)\Delta X)]\tilde{\tilde{d}}^n_{I+4}\\
&+[-2r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X)]\tilde{\tilde{d}}^n_{I+5}\\
&+[-r_1^2\Psi^{L^2}((3-0.5\alpha)\Delta X)]\tilde{\tilde{d}}^n_{I+6}\}
\end{aligned}
$$

$$
\begin{aligned}
&\tilde{\tilde{d}}^{n+1}_I-2\tilde{\tilde{d}}^n_I+\tilde{\tilde{d}}^{n-1}_I= \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(B.3)}\\
&-\frac{c^2\Delta t^2}{\alpha^2\Delta x}\{r_1^2\Psi^{L^2}((3-0.5\alpha)\Delta X)[-\tilde{\tilde{d}}^n_{I-6}+2d^n_I-\tilde{\tilde{d}}^n_{I+6}]\\
&+r_1^2\Psi^{L^2}((2-0.5\alpha)\Delta X)[-\tilde{\tilde{d}}^n_{I-4}+2d^n_I-\tilde{\tilde{d}}^n_{I+4}]\\
&+r_1^2\Psi^{L^2}((1-0.5\alpha)\Delta X)[-\tilde{\tilde{d}}^n_{I-2}+2d^n_I-\tilde{\tilde{d}}^n_{I+2}]\\
&+r_1^2\Psi^{L^2}((1+0.5\alpha)\Delta X)[-\tilde{\tilde{d}}^n_{I-2}+2d^n_I-\tilde{\tilde{d}}^n_{I+2}]\\
&+r_2^2\Psi^{SL^2}((1-0.25\alpha)\Delta x)[-\tilde{\tilde{d}}^n_{I-2}+2d^n_I-\tilde{\tilde{d}}^n_{I+2}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X)[-2d^n_{I-5}+2d^n_{I-1}+2d^n_{I+1}-2d^n_{I+5}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[-2d^n_{I-4}+2d^n_{I-2}+2d^n_{I+2}-2d^n_{I+4}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)[+2d^n_{I-4}-2d^n_{I-2}-2d^n_{I+2}+2d^n_{I+4}]\\
&+r_1r_2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2d^n_{I-4}+2d^n_{I-2}+2d^n_{I+2}-2d^n_{I+4}]\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)[+2d^n_{I-3}-2d^n_{I-1}-2d^n_{I+1}+2d^n_{I+3}]\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[-2d^n_{I-3}+2d^n_{I-1}+2d^n_{I+1}-2d^n_{I+3}]\\
&+r_1r_2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2d^n_{I-3}+2d^n_{I-1}+2d^n_{I+1}-2d^n_{I+3}]\\
&+r_1^2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[+2d^n_{I-2}-4d^n_I+2d^n_{I+2}]\\
&+r_1r_2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[+2d^n_{I-2}-4d^n_I+2d^n_{I+2}]\\
&+r_1r_2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2d^n_{I-2}+4d^n_I-2d^n_{I+2}]\}
\end{aligned}
$$

Now, by introducing Eq. (4.30), and by dividing by $\lambda^n e^{ikI\Delta x}$ we obtain

$$
\begin{aligned}
&\lambda^1 - 2 + \lambda^{-1} = \qquad\qquad (B.4)\\
&-\frac{c^2\Delta t^2}{\alpha^2\Delta x}\{r_1^2\Psi^{L^2}((3-0.5\alpha)\Delta X)[-e^{-6ik\Delta x}+2-e^{6ik\Delta x}]\\
&+r_1^2\Psi^{L^2}((2-0.5\alpha)\Delta X)[-e^{-4ik\Delta x}+2-e^{4ik\Delta x}]\\
&+r_1^2\Psi^{L^2}((1-0.5\alpha)\Delta X)[-e^{-2ik\Delta x}+2-e^{2ik\Delta x}]\\
&+r_1^2\Psi^{L^2}((1+0.5\alpha)\Delta X)[-e^{-2ik\Delta x}+2-e^{2ik\Delta x}]\\
&+r_2^2\Psi^{SL^2}((1-0.25\alpha)\Delta x)[-e^{-2ik\Delta x}+2-e^{2ik\Delta x}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X)[-2e^{-5ik\Delta x}+2e^{-ik\Delta x}+2e^{ik\Delta x}-2e^{5ik\Delta x}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[-2e^{-4ik\Delta x}+2e^{-2ik\Delta x}+2e^{2ik\Delta x}-2e^{4ik\Delta x}]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)[+2e^{-4ik\Delta x}-2e^{-2ik\Delta x}-2e^{2ik\Delta x}+2e^{4ik\Delta x}]\\
&+r_1r_2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2e^{-4ik\Delta x}+2e^{-2ik\Delta x}+2e^{2ik\Delta x}-2e^{4ik\Delta x}]\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)[+2e^{-3ik\Delta x}-2e^{-ik\Delta x}-2e^{ik\Delta x}+2e^{3ik\Delta x}]\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[-2e^{-3ik\Delta x}+2e^{-ik\Delta x}+2e^{ik\Delta x}-2e^{3ik\Delta x}]\\
&+r_1r_2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2e^{-3ik\Delta x}+2e^{-ik\Delta x}+2e^{ik\Delta x}-2e^{3ik\Delta x}]\\
&+r_1^2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[+2e^{-2ik\Delta x}-4+2e^{2ik\Delta x}]\\
&+r_1r_2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[+2e^{-2ik\Delta x}-4+2e^{2ik\Delta x}]\\
&+r_1r_2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-2e^{-2ik\Delta x}+4-2e^{2ik\Delta x}]\}
\end{aligned}
$$

$$\begin{aligned}
&\lambda^2-2\lambda+1= \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(B.5)}\\
&-\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\{r_1^2\Psi^{L^2}((3-0.5\alpha)\Delta X)[4\sin^2(3k\Delta x)]\\
&+r_1^2\Psi^{L^2}((2-0.5\alpha)\Delta X)[4\sin^2(2k\Delta x)]\\
&+r_1^2\Psi^{L^2}((1-0.5\alpha)\Delta X)[4\sin^2(k\Delta x)]\\
&+r_1^2\Psi^{L^2}((1+0.5\alpha)\Delta X)[4\sin^2(k\Delta x)]\\
&+r_2^2\Psi^{SL^2}((1-0.25\alpha)\Delta x)[4\sin^2(k\Delta x)]\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((2-0.5\alpha)\Delta X)\{4[\cos(k\Delta x)-\cos(5k\Delta x)]\}\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)\{4[\cos(2k\Delta x)-\cos(4k\Delta x)]\}\\
&+r_1^2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)\{4[\cos(4k\Delta x)-\cos(2k\Delta x)]\}\\
&+r_1r_2\Psi^{L}((3-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)\{4[\cos(2k\Delta x)-\cos(4k\Delta x)]\}\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1+0.5\alpha)\Delta X)\{4[\cos(3k\Delta x)-\cos(k\Delta x)]\}\\
&+r_1^2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)\{4[\cos(k\Delta x)-\cos(3k\Delta x)]\}\\
&+r_1r_2\Psi^{L}((2-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)\{4[\cos(3k\Delta x)-\cos(k\Delta x)]\}\\
&+r_1^2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{L}((1-0.5\alpha)\Delta X)[-8\sin^2(k\Delta x)]\\
&+r_1r_2\Psi^{L}((1+0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[-8\sin^2(k\Delta x)]\\
&+r_1r_2\Psi^{L}((1-0.5\alpha)\Delta X)\Psi^{SL}((1-0.25\alpha)\Delta x)[+8\sin^2(k\Delta x)]\}\\
&=-\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\{4[r_1\Psi^{L}((3-0.5\alpha)\Delta X)\sin(3k\Delta x)+r_1\Psi^{L}((2-0.5\alpha)\Delta X)\sin(2k\Delta x)\\
&+r_1\Psi^{L}((1-0.5\alpha)\Delta X)\sin(k\Delta x)-r_1\Psi^{L}((1+0.5\alpha)\Delta X)\sin(k\Delta x)\\
&+r_2\Psi^{SL}((1-0.25\alpha)\Delta x)\sin(k\Delta x)]^2\}\\
&=-4\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\{r_1[\Psi^{L}((3-0.5\alpha)\Delta X)\sin(3k\Delta x)+\Psi^{L}((2-0.5\alpha)\Delta X)\sin(2k\Delta x)\\
&+\Psi^{L}((1-0.5\alpha)\Delta X)\sin(k\Delta x)-\Psi^{L}((1+0.5\alpha)\Delta X)\sin(k\Delta x)]\\
&+r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)\sin(k\Delta x)]\}^2=-4\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\{r_1A^L+r_2A^{SL}\}^2\\
&=-4\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\bar{A},
\end{aligned}$$

where

$$\begin{aligned}\bar{A} = \{ & r_1[\Psi^L((3-0.5\alpha)\Delta X)\sin(3k\Delta x) \\ & + \Psi^L((2-0.5\alpha)\Delta X)\sin(2k\Delta x) \\ & + \Psi^L((1-0.5\alpha)\Delta X)\sin(k\Delta x) - \Psi^L((1+0.5\alpha)\Delta X)\sin(k\Delta x)] \\ & + r_2[\Psi^{SL}((1-0.25\alpha)\Delta x)\sin(k\Delta x)]\}^2\end{aligned} \tag{B.6}$$

and

$$\begin{aligned}A^L &= [\Psi^L((3-0.5\alpha)\Delta X)\sin(3k\Delta x) + \Psi^L((2-0.5\alpha)\Delta X)\sin(2k\Delta x) \\ & + \Psi^L((1-0.5\alpha)\Delta X)\sin(k\Delta x) - \Psi^L((1+0.5\alpha)\Delta X)\sin(k\Delta x)] \\ &= [\Psi^L((3-0.5\alpha)\Delta X)\sin(3\bar{k}\Delta X) + \Psi^L((2-0.5\alpha)\Delta X)\sin(2\bar{k}\Delta X) \\ & + \Psi^L((1-0.5\alpha)\Delta X)\sin(\bar{k}\Delta X) - \Psi^L((1+0.5\alpha)\Delta X)\sin(\bar{k}\Delta X),\end{aligned}$$

with $\bar{k} = 2k$, and

$$A^{SL} = [\Psi^{SL}((1-0.25\alpha)\Delta x)\sin(k\Delta x)].$$

Therefore, we can write

$$\lambda^2 - 2\lambda + 1 = -4\frac{c^2\Delta t^2}{\alpha^2\Delta x}\lambda\bar{A}, \tag{B.7}$$

$$\lambda^2 - 2(1 - 2\frac{c^2\Delta t^2}{\alpha^2\Delta x}\bar{A})\lambda + 1 = 0, \tag{B.8}$$

and can find

$$\lambda_{1,2} = 1 - 2\frac{c^2\Delta t^2}{\alpha^2\Delta x}\bar{A} \pm \sqrt{(1 - 2\frac{c^2\Delta t^2}{\alpha^2\Delta x}\bar{A})^2 - 1}. \tag{B.9}$$

For stability, we require

$$|\lambda_{1,2}| \leq 1, \tag{B.10}$$

which is satisfied for $\bar{A} = 0$ or, with $\bar{A} \neq 0$, if

$$(1 - 2\frac{c^2\Delta t^2}{\alpha^2\Delta x}\bar{A})^2 \leq 1, \tag{B.11}$$

$$2\frac{c^2\Delta t^2}{\alpha^2\Delta x}\bar{A} \leq 2, \tag{B.12}$$

$$c^2\Delta t^2 \leq \frac{\alpha^2\Delta x}{\bar{A}}. \tag{B.13}$$

Now, we show that

$$\begin{aligned}\bar{A} &= (r_1 A^L + r_2 A^{SL})^2 \\ &= \left[(1-r)A^L + rA^{SL}\right]^2 \leq \max({A^L}^2, {A^{SL}}^2).\end{aligned} \tag{B.14}$$

In order to consider all possibilities with respect to the signs of A^L and A^{SL}, we recast from Eq. (B.14) as

$$\bar{A} = \left[(1-r)|A^L| - r|A^{SL}|\right]^2. \tag{B.15}$$

We start by considering the case for which $|A^{SL}| \geq |A^{L}|$, so we want

$$\bar{A} = \left[(1-r)|A^{L}| - r|A^{SL}|\right]^{2} \leq |A^{SL}|^{2}, \tag{B.16}$$

$$-|A^{SL}| \leq (1-r)|A^{L}| - r|A^{SL}| \leq |A^{SL}|. \tag{B.17}$$

Recalling that $0 \leq r \leq 1$, we show the left-hand side of Eq. (B.17) as follows

$$-|A^{SL}| \leq -r|A^{SL}| \leq (1-r)|A^{L}| - r|A^{SL}|, \tag{B.18}$$

and the right-hand side of Eq. (B.17) as follows

$$(1-r)|A^{L}| - r|A^{SL}| \leq (1-r)|A^{L}| \leq (1-r)|A^{SL}| \leq |A^{SL}|. \tag{B.19}$$

Similarly, if $|A^{L}| \geq |A^{SL}|$, we show

$$-|A^{L}| \leq (1-r)|A^{L}| - r|A^{SL}| \leq |A^{L}|. \tag{B.20}$$

The left-hand side of Eq. (B.20) can be shown as follows:

$$-|A^{L}| \leq (1-r)|A^{L}| - r|A^{SL}|, \tag{B.21}$$

$$|A^{SL}| \leq \frac{2-r}{r}|A^{L}|, \tag{B.22}$$

Since $\frac{2-r}{r} \geq 0$ and $|A^{L}| \geq |A^{SL}|$, Eq. (B.22) is always satisfied. Now, for the right-hand side of

Eq. (B.20):

$$(1-r)|A^L| - r|A^{SL}| \leq |A^L|, \tag{B.23}$$

$$-r|A^{SL}| \leq r|A^L|, \tag{B.24}$$

$$|A^{SL}| \leq |A^L|, \tag{B.25}$$

which is also always satisfied.

Chapter 5

A reproducing kernel enhanced approach for peridynamic solutions

5.1 The Peridynamic Theory

The peridynamic theory [18][24][111][116][113] is a nonlocal reformulation of the classical theory of continuum mechanics. The peridynamic theory is nonlocal since, given a bounded body $\Omega \subset \mathbb{R}^d$ with d being the spatial dimension ($d = 1,2,3$), a material point $\mathbf{x} \in \Omega$ interacts with all points $\mathbf{x}' \in \mathbb{R}^d$ within a finite distance δ called horizon, which represents the maximum distance for nonlocal interaction in the material model, and defines the neighborhood of $\mathbf{x}$:

$$\mathscr{H}_{\mathbf{x}} := \{\mathbf{x}' \in \mathbb{R}^d | \|\boldsymbol{\xi}\| \leq \delta\}, \tag{5.1}$$

where $\boldsymbol{\xi} := \mathbf{x}' - \mathbf{x}$ is called the bond connecting the material point $\mathbf{x}$ to the material point $\mathbf{x}'$ in the reference configuration. In this paper, it will be assumed that the horizon is independent of $\mathbf{x}$. The material points contained in the neighborhood $\mathscr{H}_{\mathbf{x}}$ are referred to as the family of $\mathbf{x}$.

By replacing the stress divergence in the momentum equation with a volume integral, the peridynamic theory extends the classical theory of continuum mechanics to allow for discontinuities in the displacement field. The peridynamic equation of motion for a material point $\mathbf{x} \in \Omega$ at time $t \geq 0$ is

$$\rho(\mathbf{x}) \frac{\partial^2 \mathbf{u}}{\partial t^2}(\mathbf{x}, t) = \int_{\mathscr{H}_{\mathbf{x}}} \hat{\mathbf{f}}(\mathbf{x}, \mathbf{x}', t) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t), \tag{5.2}$$

where $\hat{\mathbf{f}}(\mathbf{x}, \mathbf{x}', t)$ is the force density (with units of force per volume squared) that the material point $\mathbf{x}'$ exerts on the material point $\mathbf{x}$, $\mathbf{b}$ is a prescribed body force density field, ρ is the mass density, and $\mathbf{u}$ represents the displacement field. In general, the force density can be expressed as

$$\hat{\mathbf{f}}(\mathbf{x}, \mathbf{x}', t) = \underline{\mathbf{T}}[\mathbf{x}, t]\langle \boldsymbol{\xi} \rangle - \underline{\mathbf{T}}[\mathbf{x}', t]\langle -\boldsymbol{\xi} \rangle, \tag{5.3}$$

where $\underline{\mathbf{T}}$ is an operator called force state [113] that maps the bond $\boldsymbol{\xi}$ to a force density (force per volume squared). The spatial dependence of the force state on $\mathbf{x}$ and t is written in square brackets, while the bond the operator acts on is contained within angle brackets. The expression in Eq. (5.3) satisfies $\hat{\mathbf{f}}(\mathbf{x},\mathbf{x}',t) = -\hat{\mathbf{f}}(\mathbf{x}',\mathbf{x},t)$, which guarantees conservation of linear momentum [24][116][113]. The force state $\underline{\mathbf{T}}$ contains the information regarding the material constitutive behavior and, depending on its definition, the peridynamic material models fall into one of two categories: bond-based or state-based [113]. State-based material models can be further subdivided into ordinary state-based and non-ordinary state-based [53]. In ordinary state-based material models, force densities act in the direction of the corresponding bond in the deformed configuration: $\mathbf{y}(\mathbf{x}',t) - \mathbf{y}(\mathbf{x},t) = \boldsymbol{\xi} + \boldsymbol{\eta}$, where $\mathbf{y}(\mathbf{x},t) = \mathbf{x} + \mathbf{u}(\mathbf{x},t)$ is the deformed position of $\mathbf{x}$ at time t and $\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}',t) - \mathbf{u}(\mathbf{x},t)$ is the relative displacement between $\mathbf{x}$ and $\mathbf{x}'$ at time t. In non-ordinary state-based material models, force densities are not restricted to act in the deformed bond direction. Bond-based models are the simplest class of material models in peridynamics [111]. In bond-based models, the (pairwise) force density depends only on the deformation of the bond $\boldsymbol{\xi}$, meaning that $\underline{\mathbf{T}}[\mathbf{x},t]\langle\boldsymbol{\xi}\rangle$ is such that $\hat{\mathbf{f}}(\mathbf{x},\mathbf{x}',t) = \mathbf{f}(\mathbf{u}(\mathbf{x}',t) - \mathbf{u}(\mathbf{x},t), \mathbf{x}' - \mathbf{x}, t)$ [113]. For a static problem, the peridynamic equilibrium equation for $\mathbf{x} \in \Omega$ is

$$-\int_{\mathscr{H}_{\mathbf{x}}} \underline{\mathbf{T}}[\mathbf{x}]\langle\boldsymbol{\xi}\rangle - \underline{\mathbf{T}}[\mathbf{x}']\langle-\boldsymbol{\xi}\rangle dV_{\mathbf{x}'} = \mathbf{b}(\mathbf{x}). \tag{5.4}$$

Different from the classical local problems based on PDEs, where boundary conditions are imposed over a lower-dimensional domain $\partial\Omega \subset \mathbb{R}^{d-1}$, in peridynamics boundary conditions are generally imposed over a nonzero volumetric layer $\mathscr{B}\Omega \subset \mathbb{R}^d$ surrounding the domain of interest $\Omega \subset \mathbb{R}^d$ due to the integral nature of the equilibrium equation. This leads to an overall problem domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. Such imposition of boundary conditions guarantees well-posedness of the resulting nonlocal boundary-value problem [101][73][126]. The width of $\mathscr{B}\Omega$ is typically taken to be at least equal to the horizon δ [4][24][93] (typically δ for bond-based models and 2δ for state-based models).

5.1.1 Node-based meshfree discretization of peridynamic equations

The continuum peridynamic equation (5.2) can be discretized with different approaches. Due to the simplicity of its implementation and its relatively low computational cost, compared to other discretizations, the most common discretization approach in peridynamics is the node-based discretization method of [112].

Consider a domain $\overline{\overline{\Omega}}$, and let it be discretized by a set of nodes $\mathbf{x}_I \in \overline{\overline{\Omega}}$, each associated with a material cell Ω_I such that $\bigcup_I \Omega_I = \overline{\overline{\Omega}}$ and $\Omega_I \cap \Omega_J = \varnothing$ for $I \neq J$. Each cell Ω_I has a volume V_I. In this approach, the solution at each node is considered as representative of the solution in its associated cell. Following [93], the peridynamic equation of motion is first integrated over Ω_I, leading to

$$\int_{\Omega_I} \rho(\mathbf{x}) \frac{\partial^2 \mathbf{u}}{\partial t^2}(\mathbf{x},t) dV_{\mathbf{x}} = \int_{\Omega_I} \int_{\mathscr{H}_{\mathbf{x}}} \underline{\mathbf{T}}[\mathbf{x},t]\langle \mathbf{x}' - \mathbf{x} \rangle - \underline{\mathbf{T}}[\mathbf{x}',t]\langle \mathbf{x} - \mathbf{x}' \rangle dV_{\mathbf{x}'} dV_{\mathbf{x}} + \int_{\Omega_I} \mathbf{b}(\mathbf{x},t) dV_{\mathbf{x}}. \quad (5.5)$$

Because of the antisymmetry of the integrand in the first term of the right-hand side of Eq. (5.5), the self interaction of the cell Ω_I vanishes [93]. By employing a one-point quadrature rule for the outer integral over Ω_I with $\mathbf{x}_I \in \Omega$ as quadrature point and V_I as quadrature weight, and by subsequently dividing by V_I, Eq. (5.5) becomes

$$\rho(\mathbf{x}_I) \frac{\partial^2 \mathbf{u}}{\partial t^2}(\mathbf{x}_I,t) = \int_{\mathscr{H}_{\mathbf{x}} \setminus \Omega_I} \underline{\mathbf{T}}[\mathbf{x}_I,t]\langle \mathbf{x}' - \mathbf{x}_I \rangle - \underline{\mathbf{T}}[\mathbf{x}',t]\langle \mathbf{x}_I - \mathbf{x}' \rangle dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}_I,t). \quad (5.6)$$

Now, take the discrete family of $\mathbf{x}_I \in \Omega$ for the node-based approach as

$$\mathcal{F}_I = \left\{ \mathbf{x}_J \in \overline{\overline{\Omega}}, J \neq I | \|\mathbf{x}_J - \mathbf{x}_I\| \leq \delta \right\}, \quad (5.7)$$

and express the integral in Eq. (5.6) as a summation of integrals over the cells of the family nodes.

Then, we have

$$\rho(\mathbf{x}_I)\frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x}_I,t)=\sum_{J\in\mathcal{F}_I}\left(\int_{\Omega_J}\underline{\mathbf{T}}[\mathbf{x}_I,t]\langle\mathbf{x}'-\mathbf{x}_I\rangle-\underline{\mathbf{T}}[\mathbf{x}',t]\langle\mathbf{x}_I-\mathbf{x}'\rangle dV_{\mathbf{x}'}\right)+\mathbf{b}(\mathbf{x}_I,t). \tag{5.8}$$

Finally, each integral in the summation is evaluated using a one-point quadrature rule over the J-th cell with $\mathbf{x}_J$ and V_J the quadrature point and weight, respectively. The node-based meshfree discrete peridynamic equation is then

$$\rho(\mathbf{x}_I)\frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x}_I,t)=\sum_{J\in\mathcal{F}_I}(\underline{\mathbf{T}}[\mathbf{x}_I,t]\langle\mathbf{x}_J-\mathbf{x}_I\rangle-\underline{\mathbf{T}}[\mathbf{x}_J,t]\langle\mathbf{x}_I-\mathbf{x}_J\rangle)V_J+\mathbf{b}(\mathbf{x}_I,t). \tag{5.9}$$

The definition of $\mathcal{F}_I$ in Eq. (5.7) was introduced in [93] in conjunction with a Full Volume (FV) approach. In the FV approach, if a neighboring node is within the neighborhood of $\mathbf{x}_I$, its full volume is used as quadrature weight regardless of how much of it is covered by the neighborhood of $\mathbf{x}_I$. To improve the accuracy of the integration, different definitions of $\mathcal{F}_I$ [26] and integration weights [85][93] as well as the use of additional integration points [92] have been proposed.

5.2 Reproducing Kernel Enhanced Peridynamics

The node-based meshfree discretization as discussed in Section 5.1.1, while simple and computationally efficient compared to other discretization approaches such as finite elements, presents some drawbacks. First, the use of a nodal integration approach can strongly affect accuracy and convergence rates, particularly for non-uniform discretizations [7][25]. This approach considers a piecewise-constant approximation of the displacement field and limits the convergence rate of numerical solutions to first order. To obtain higher convergence rates, other solution approaches, such as the use of finite difference or finite element discretizations with linear basis functions have been proposed [7][100][141][140].

The proposed RK approach introduces the discrete RK approximation $\mathbf{u}^h(\mathbf{x})$ of the displacement field $\mathbf{u}(\mathbf{x})$ (Eq. (3.3)) into the peridynamic governing equation (Eq. (5.2)). As presented in Chapter 3, the RK approximation can be constructed with arbitrary smoothness and completeness, leading to an increased convergence rate in a meshfree framework. The RK enhanced approach is here presented for bond-based peridynamic models, i.e., $\hat{\mathbf{f}}(\mathbf{x},\mathbf{x}',t) = \mathbf{f}(\mathbf{u}(\mathbf{x}',t)-\mathbf{u}(\mathbf{x},t),\mathbf{x}'-\mathbf{x},t)$. Furthermore, in this work, numerical integration is performed by means of a background grid of Gauss integration points (see, for example, Figure 5.1). This approach is thereon referred to as Gauss integration scheme 1.

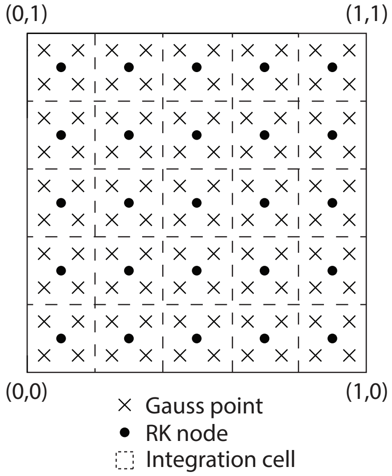


Figure 5.1: Meshfree discretization of a two-dimensional unit square domain with background integration grid.

In [81], another approach (Gauss integration scheme 2) is also employed, where Gauss integration points are placed inside the neighborhood of each node as shown in Figure 5.2 and Figure 5.3, representing a 1D and a 2D case, respectively. In 1D, Gauss integration points are distributed in a [-1,1] interval and mapped into the neighborhood of a point x, $[x-\delta, x+\delta]$. In 2D, the neighborhood of a point $\mathbf{x}$ is a disk of radius δ. Any integral I on this circle can be transformed using polar coordinates as follows:

$$\begin{aligned} I &= \int_{-\delta}^{\delta}\int_{-\sqrt{\delta^2-x^2}}^{\sqrt{\delta^2-x^2}} f(x,y)dydx \\ &= \int_0^{\delta}\int_0^{2\pi} f(r\cos\theta, r\sin\theta)rd\theta dr \\ &= \int_0^1\int_0^1 f(\delta\xi\cos(2\pi\eta), \delta\xi\sin(2\pi\eta))\delta\xi 2\pi d\eta d\xi. \end{aligned} \tag{5.10}$$

The idea is to place the Gauss integration points in a unit square $[0,1]\times[0,1]$ and then transform their coordinates to polar coordinates by a change of variables so that they fall in $[0,\delta]\times[0,2\pi]$. In this way, all integration points are contained within each point neighborhood regardless of the size of the horizon δ. This change of coordinates results in a non-uniform angular distribution of Gauss quadrature points in the neighborhood (as in Figure 5.3a). Another technique [36] [105] [118] that creates uniform angular distributions of quadrature points is also adopted for the so-called Gauss integration scheme 2 (as in Figure 5.3b).

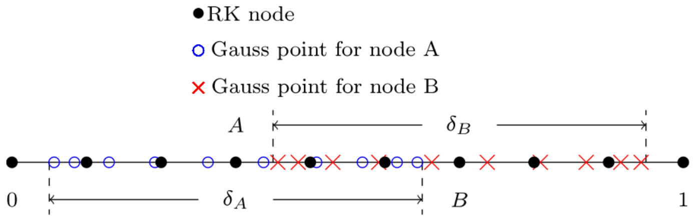
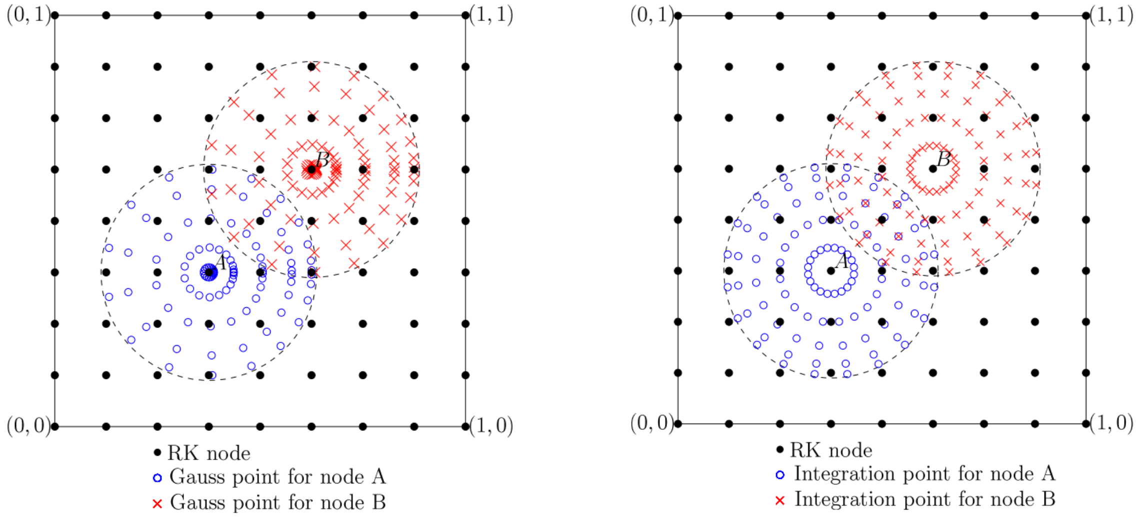


Figure 5.2: Meshfree discretization of a 1D domain with Gauss integration points within the neighborhood of each node.



(a) Non-uniform Gauss Points.

(b) Uniform integration points in angular direction.

Figure 5.3: Meshfree discretization of a two-dimensional unit square domain with integration points within the neighborhood of each node. The dashed lines represent the limits of the neighborhoods of nodes A and B.

Now, the reproducing kernel enhanced approach is derived by introducing the RK approximation in the field variables. The Gauss integration scheme is used to evaluate the integral in the peridynamic equilibrium equation. The peridynamic equation of motion is first evaluated at node $\mathbf{x}_I \in \Omega$:

$$\rho(\mathbf{x}_I)\frac{\partial^2 \mathbf{u}}{\partial t^2}(\mathbf{x}_I,t) = \int_{\mathcal{H}_{\mathbf{x}}} \mathbf{f}(\mathbf{u}(\mathbf{x}',t) - \mathbf{u}(\mathbf{x}_I,t), \mathbf{x}' - \mathbf{x}_I, t)\, dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}_I,t). \tag{5.11}$$

The integral in Eq. (5.11) is then evaluated as a summation over the integration points $\mathbf{x}_g \in \mathcal{F}_I$ with integration weights w_g:

$$\rho(\mathbf{x}_I)\frac{\partial^2 \mathbf{u}}{\partial t^2}(\mathbf{x}_I,t) = \sum_{g:\mathbf{x}_g \in \mathcal{F}_I} \mathbf{f}(\mathbf{u}(\mathbf{x}_g,t) - \mathbf{u}(\mathbf{x}_I,t), \mathbf{x}_g - \mathbf{x}_I, t)\, w_g + \mathbf{b}(\mathbf{x}_I,t), \tag{5.12}$$

where the discrete family of $\mathbf{x}_I$, $\mathcal{F}_I$, is now

$$\mathcal{F}_I = \left\{ \mathbf{x}_g \in \overline{\overline{\Omega}} | \|\mathbf{x}_g - \mathbf{x}_I\| \leq \delta \right\}. \tag{5.13}$$

It can be noted that, if the Gauss integration scheme 1 is used, the node-based scheme presented in Section 5.1.1 is recovered for a choice of quadrature point $\mathbf{x}_J$ and quadrature weight V_J. The displacement values at $\mathbf{x}_I$ and $\mathbf{x}_g$ are then computed by their RK approximation (*cf.* Eq. (3.3)), leading to

$$\begin{aligned} \rho(\mathbf{x}_I) \sum_{J=1}^{NP} \Psi_J(\mathbf{x}_I) \frac{d^2 \mathbf{u}_J(t)}{dt^2} = & \sum_{g:\mathbf{x}_g \in \mathcal{F}_I} \mathbf{f}\left(\sum_{J=1}^{NP} (\Psi_J(\mathbf{x}_g) - \Psi_J(\mathbf{x}_I)) \mathbf{u}_J(t), \mathbf{x}_g - \mathbf{x}_I, t \right) w_g \\ & + \mathbf{b}(\mathbf{x}_I, t). \end{aligned} \tag{5.14}$$

Equation (5.14) can then be solved to obtain the nodal coefficients, after providing initial and boundary conditions.

In the case of a one-dimensional static linear problem so that, for example, $f(u(x',t) - u(x,t), x' - x, t) = c(u(x') - u(x))$, where c is the material micromodulus, Eq. (5.14) becomes

$$- \sum_{g:x_g \in \mathcal{F}_I} \sum_{J=1}^{NP} c(\Psi_J(x_g) - \Psi_J(x_I)) u_J w_g = b(x_I) \quad \forall x_I \in \Omega, \tag{5.15}$$

which can be rewritten as

$$\mathbf{A}_I \mathbf{u} = b_I \quad \forall x_I \in \Omega, \tag{5.16}$$

where

$$\mathbf{A}_I^T = \begin{bmatrix} -\sum_{g:x_g \in \mathcal{F}_I} c(\Psi_1(x_g) - \Psi_1(x_I)) w_g \\ -\sum_{g:x_g \in \mathcal{F}_I} c(\Psi_2(x_g) - \Psi_2(x_I)) w_g \\ \cdots \\ -\sum_{g:x_g \in \mathcal{F}_I} c(\Psi_{NP}(x_g) - \Psi_{NP}(x_I)) w_g \end{bmatrix}, \tag{5.17}$$

$$\mathbf{u}^T = \begin{bmatrix} u_1 & u_2 & \dots & u_{NP} \end{bmatrix}, \tag{5.18}$$

and

$$b_I = b(x_I). \tag{5.19}$$

The discrete equations in (5.16) for all the interior nodes can be combined to yield

$$\mathbf{Au} = \mathbf{b}, \tag{5.20}$$

where

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{A}_1 & \dots & \mathbf{A}_I & \dots & \mathbf{A}_{n_i} \end{bmatrix}, \tag{5.21}$$

$$\mathbf{b}^T = \begin{bmatrix} b_1 & \dots & b_I & \dots & b_{n_i} \end{bmatrix}, \tag{5.22}$$

and n_i denotes the number of nodes $x_I \in \Omega$. Equation (5.15) holds for all $x_I \in \Omega$. For points $x_K \in \mathscr{B}\Omega$, the equations associated with the prescribed boundary conditions hold. Now, it has to be noted that The RK approximation coefficients (usually called generalized coefficients) are not equivalent to the approximate function values at the nodes (i.e., $u_I \neq u^h(\mathbf{x}_I)$). In other words, the RK shape functions, different from the FEM ones, lack the Kronecker delta property. Because of this, imposition of essential boundary conditions in the RKPM framework is done differently from that in FEM. Several methods have been developed for the enforcement of Dirichlet boundary conditions in the RK approximation framework [29][62][64]. In the static problems considered in this work, we limit ourselves to Dirichlet boundary conditions and impose them strongly by collocation [29][60][56][64] at the boundary nodes:

$$u^h(x_K) = \sum_{J=1}^{NP} \Psi_J(x_K) u_J = g_K \quad \forall x_K \in \mathscr{B}\Omega \tag{5.23}$$

or

$$\mathbf{G}_K \mathbf{u} = g_K \quad \forall x_K \in \mathscr{B}\Omega, \tag{5.24}$$

where g_k is the prescribed boundary value and

$$\mathbf{G}_K = \begin{bmatrix} \Psi_1(x_K) & \Psi_2(x_K) & \ldots & \Psi_{NP}(x_K) \end{bmatrix}. \tag{5.25}$$

The discrete equations in Eq. (5.24) for all Dirichlet boundary nodes can be combined to yield

$$\mathbf{Gu} = \mathbf{g}, \tag{5.26}$$

where

$$\mathbf{G}^T = \begin{bmatrix} \mathbf{G}_1 & \ldots & \mathbf{G}_K & \ldots & \mathbf{G}_{n_k} \end{bmatrix}, \tag{5.27}$$

$$\mathbf{g}^T = \begin{bmatrix} g_1 & \ldots & g_K & \ldots & g_{n_k} \end{bmatrix}, \tag{5.28}$$

and n_k denotes the number of nodes $x_K \in \mathscr{B}\Omega$. By combining Eq. (5.15) and Eq. (5.23), and by considering every $x_I \in \overline{\overline{\Omega}}$, the following matrix equation is obtained:

$$\mathbf{Ku} = \mathbf{f}, \tag{5.29}$$

where

$$\mathbf{K}^T = \begin{bmatrix} \mathbf{A}_1 & \ldots & \mathbf{A}_{n_i} & \mathbf{G}_1 & \ldots & \mathbf{G}_{n_k} \end{bmatrix}, \tag{5.30}$$

$$\mathbf{f}^T = \begin{bmatrix} b_1 & \ldots & b_{n_i} & g_1 & \ldots & g_{n_k} \end{bmatrix}. \tag{5.31}$$

Eq. (5.29) can be solved to obtain the generalized coefficients vector $\mathbf{u}$. A similar procedure can be followed in higher dimensions.

5.3 Numerical Results

5.3.1 One-dimensional static peridynamic problem

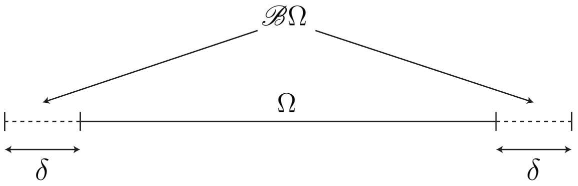


Figure 5.4: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$.

Following [93], consider a one-dimensional unit length domain (see Figure 7.8) $\overline{\overline{\Omega}} = [0,1]$, with inner domain $\Omega = (\delta, 1-\delta)$ and boundary layer $\mathscr{B}\Omega = [0,\delta] \cup [1-\delta, 1]$, and a one-dimensional force state

$$\underline{\mathrm{T}}[x]\langle\xi\rangle = \frac{1}{2} c w(|\xi|)(u(x') - u(x)), \tag{5.32}$$

where $\xi = x' - x$, $w(|\xi|)$ is an influence function, and c is a constitutive constant. An influence function is a scalar-valued function, with a finite support determined by the horizon, used to weight the force state. It is commonly dependent on the bond length (spherical influence function) [95][113] and it is zero for bond lengths greater than the horizon. In this work the influence function is taken as follows:

$$w(|\xi|) = \begin{cases} \frac{1}{|\xi|^{\alpha}}, & \text{for } |\xi| \leq \delta \\ 0, & \text{for } |\xi| > \delta \end{cases}, \tag{5.33}$$

where $\alpha = 0, 1$. The constitutive constant c is taken as [93]

$$c = \frac{2K}{m}, \tag{5.34}$$

where K is a constant and m is the so-called weighted volume [18][24][95][113]:

$$m = \int_{-\delta}^{\delta} w(|\xi|)\xi^2 d\xi. \tag{5.35}$$

The static linear bond-based peridynamic problem associated with Eq. (6.80) is

$$\begin{cases} -\int_{x-\delta}^{x+\delta} cw(|\xi|)(u(x') - u(x))dx' = b(x), & \text{for } x \in \Omega \\ u(x) = g(x), & \text{for } x \in \mathscr{B}\Omega \end{cases} \tag{5.36}$$

where $g(x)$ is a prescribed Dirichlet boundary condition function. To assess the performance of the RK enhanced approach and to perform convergence studies of the corresponding numerical solutions, the method of manufactured solutions is employed. An analytical solution $u(x)$ to the boundary value problem in Eq. (5.36) is chosen. Based on such solution, the boundary conditions are taken so that $g(x) = u(x)$ $\forall x \in \mathscr{B}\Omega$ and the associated body force density field $b(x)$ in Ω is computed.

Three manufactured solutions are considered: a quadratic polynomial, a cubic polynomial, and an exponential function. For the polynomial cases, we thus have the general form $u(x) = k_3x^3 + k_2x^2 + k_1x + k_0$ with k_3, k_2, k_1, and k_0 being constant coefficients. The body force density for Eq. (5.36) is then

$$\begin{aligned} b(x) &= -\int_{x-\delta}^{x+\delta} cw(|\xi|)[k_3(x'^3 - x^3) + k_2(x'^2 - x^2) + k_1(x' - x)]dx' \\ &= -\int_{-\delta}^{\delta} cw(|\xi|)[k_3((x+\xi)^3 - x^3) + k_2((x+\xi)^2 - x^2) + k_1\xi]d\xi \\ &= -2K(3k_3x + k_2), \end{aligned} \tag{5.37}$$

where we used Eq. (6.83).

Similarly, for the manufactured exponential solution, the corresponding body force density field is

$$b(x) = -\int_{x-\delta}^{x+\delta} cw(|\xi|)[e^{x'} - e^{x}]dx' = 2c\int_{0}^{\delta} cw(|\xi|)[1-\cosh(\xi)]d\xi e^{x}. \tag{5.38}$$

For $\alpha = 0$ in Eq. (5.33),

$$b(x) = 2c[\delta - \sinh(\delta)]e^{x} = \frac{4K}{m}[\delta - \sinh(\delta)]e^{x}, \tag{5.39}$$

while for $\alpha = 1$

$$b(x) = 2c[\gamma - \mathrm{Chi}(\delta) + \ln(\delta)]e^{x} = \frac{4K}{m}[\gamma - \mathrm{Chi}(\delta) + \ln(\delta)]e^{x}, \tag{5.40}$$

where γ is the Euler-Mascheroni constant and the hyperbolic cosine integral is

$$\mathrm{Chi}(z) := \gamma + \ln(z) + \int_{0}^{z} \frac{\cosh(x) - 1}{x} dx. \tag{5.41}$$

Convergence in one dimension with h-refinement

The concept of convergence in peridynamics [24][25] can refer either to: 1) the convergence of a nonlocal peridynamic model (or solution) to a local (PDE) counterpart as the horizon δ goes to zero, under proper regularity assumptions; or 2) the convergence of a discrete numerical solution to the continuum nonlocal solution of a peridynamic problem as the discretization becomes finer (i.e., the number of discretization nodes N increases) while the horizon δ is kept fixed (N-convergence). In this work, we focus on N-convergence. To perform convergence studies, the error L^2 norm of the numerical solution is computed by using Gauss integration points in each interval between adjacent nodes:

$$\|u^h - u\|_2 = \left[\int_\Omega (u^h(x) - u(x))^2 dx\right]^{\frac{1}{2}} = \left[\sum_{n_c=1}^{N-1} \sum_{g=1}^{n_g} (u^h(x_g^{n_c}) - u(x_g^{n_c}))^2 w_g^{n_c}\right]^{\frac{1}{2}}, \tag{5.42}$$

where the outer summation is over the number of intervals between adjacent nodes while $\{x_g^{n_c}\}_{g=1,\ldots,n_g}$ and $\{w_g^{n_c}\}_{g=1,\ldots,n_g}$ are, respectively, the Gauss points and associated weights in each interval. As discussed in Section 5.2, a piecewise-constant approximation is considered for the node-based approach, while the RK approximation is used for the RK enhanced scheme.

In [93] the convergence behavior of a node-based solution to the problem described by Eq. (5.36) was studied and integration error was identified as one of the main contributors to an oscillatory behavior of the error convergence results (see Appendix B). For this reason, we opt to employ Gauss integration based on the schemes described in Section 5.2. Moreover, when Gauss points are placed in cells coinciding with nodal intervals as in Gauss integration scheme 1, the refinement in discretization is performed so that $\delta/h = k$ with $k \in \mathbb{N}$ and h is the nodal spacing in a uniform discretization. In this way the integration cells are fully contained in the neighborhood of each discretization node (see Figure 5.5). For consistency, the same refinement scheme is also used for Gauss integration scheme 2.

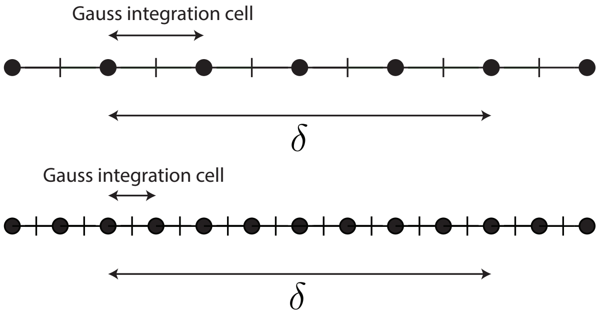


Figure 5.5: *h*-refinement scheme

Furthermore, in this work, the node-based approach discussed in Section 5.1.1 is used for comparison. However, in the node-based scheme used in this paper, the cells Ω_J of the nodes x_J near the boundary of the neighborhood are only partially included in the neighborhood. For this reason, the integration weights (i.e., the material cells' volumes V_J) for the node-based approach were computed by using the so-called Partial Volume (PV) algorithm [85][93]. This algorithm defines the family of x_I as in Eq. (5.7) but modifies the cell volumes as follows:

$$V_J = \begin{cases} \frac{1}{h}\left[\delta - \left(|x_J - x_I| - \frac{h}{2}\right)\right]\bar{V}_J & \text{if } |x_J - x_I| \geq \delta - \frac{h}{2} \\ \\ \bar{V}_J & \text{otherwise} \end{cases} \tag{5.43}$$

where $\bar{V}_J$ is the full cell volume.

Convergence of numerical solutions with Gauss integration in one dimension

In this section, the convergence of the numerical solutions to the exact solution of the peridynamic problem in Eq. (5.36) with Gauss integration is investigated. We consider first the case with a manufactured quadratic solution $u(x) = x^2$. For this problem, we choose $\alpha = 0$ in the influence function, $K = 1$, $\delta = 0.04$, $b(x) = -2$, and $g(x) = x^2$. The weighted volume m is computed analytically according to Eq. (6.84).

We consider the Gauss integration scheme 1 with Gauss integration points placed in the intervals between nodes. Convergence is studied by solving the problem for an increasing number of uniformly spaced nodes through the procedure outlined in Section 5.3.1: we start with 76 nodes ($\delta/h = 3$) and increase the number of nodes by 25 until we reach 351 nodes ($\delta/h = 14$). Under this refinement, whole integration cells and all the Gauss points in them are contained in the neighborhood thus reducing the integration error. Two numerical solution schemes are used: the node-based method discussed in Section 5.3.1 with partial volumes and the enhanced RK approximation approach with Gauss quadrature scheme 1 (*cf.* Section 5.2). For the RK formulation, a first-order basis, a cubic B-spline kernel, and a support size $a = 1.001h$ were used.

Integration was performed by using two Gauss points in each interval between adjacent nodes, as it was observed that for this type of refinement, increasing further the number of Gauss points did not result in a different convergence behavior. The error L^2 norm is computed by using three Gauss points in each integration cell. Convergence results for the two methods are presented in Figure 5.6. The average convergence rates r, for a linear fit of the whole data (in a least-squares sense) is also shown in Figure 5.6.

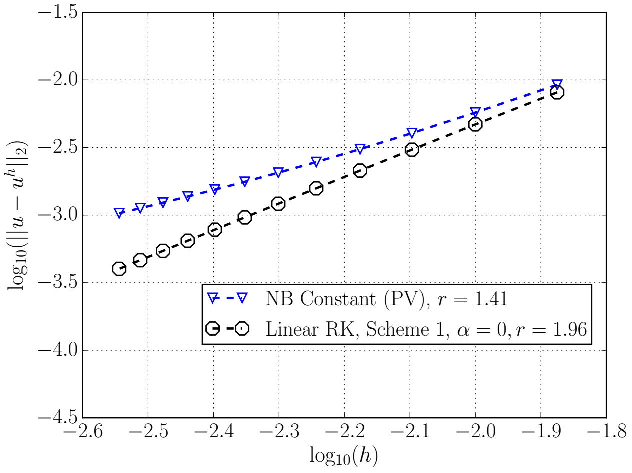


Figure 5.6: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$, using the node-based meshfree approximation with partial volumes (NB Constant (PV)) and linear RK approximation with $a = 1.001h$.

Asymptotically, i.e., if the convergence rate is computed using only the last three data points, the node-based method with partial volumes exhibits a convergence rate close to first order. Using the linear RK approximation increases the convergence rate to second order.

In [81], the same problem was also solved using the Gauss integration scheme 2 presented in Section 5.2. A total of 1000 Gauss integration points within each neighborhood were used so to reduce the effect of integration error on the convergence behavior. Values of $\alpha = 0$ and $\alpha = 1$ in the peridynamic influence function are considered and three Gauss integration points per background integration cell were used to compute the error L^2 norm. A linear RK approximation and support size $a = 1.001h$ were chosen. A similar behaviour as the one presented in Figure 5.6 for Gauss integration scheme 1 was observed: the same convergence rate of 1.96 is reported for $\alpha = 0$, while a slightly lower one ($r = 1.71$) is observed for $\alpha = 1$.

Let us now consider the problem outlined in Eq. (5.36) for a manufactured cubic solution $u(x) = x^3$. For this case, $\delta = 0.04$, $b(x) = -6x$, $g(x) = x^3$, and $K = 1$. The problem is solved using an increasing number of uniformly spaced nodes from 76 ($\delta/h = 3$) to 351 ($\delta/h = 14$) in increments of 25. Both the node-based method with partial volumes and the RK enhanced approach are used. The RK formulation is employed both with first-order and second-order bases. In the first case, the support size is $a = 1.001h$, while in the latter $a = 2.1h$. In both cases a cubic B-spline kernel is employed. First, $\alpha = 0$ has been chosen in the influence function and the Gauss integration scheme 1 with 4 Gauss points in each interval between adjacent nodes is used. Three Gauss integration points per background integration cell were used to compute the error L^2 norm. Figure 5.7 shows the error convergence results for the two methods as well as the average convergence rates r, obtained by a least-squares fit of all the error points.

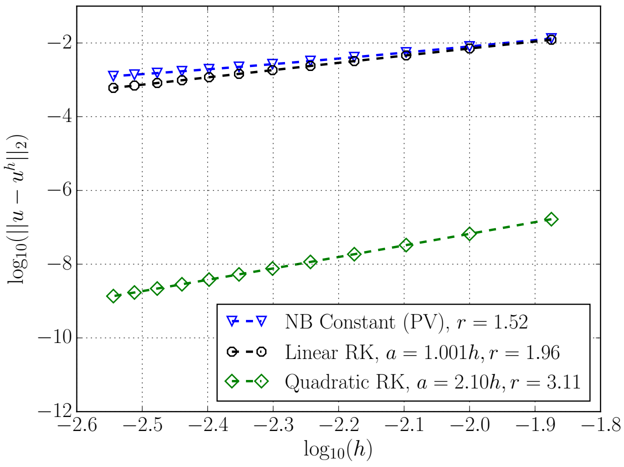


Figure 5.7: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^3$ using node-based meshfree with PV (NB Constant (PV)) and linear and quadratic RK approximations for $\alpha = 0$. Gauss integration scheme 1 is used.

Similar to the case with manufactured quadratic solution, the node-based method with partial volumes exhibits a convergence rate that is asymptotically close to linear, while the linear RK approximation increases it to second order. Raising the basis order in the RK approximation to quadratic further increases the convergence rate to close to third order. Similar results were obtained using the Gauss integration scheme 2 in [81].

Lastly, a problem with manufactured exponential solution is considered. The following data are used: $\delta = 0.04$, $\alpha = 0$, $K = 1$, $b(x) = \frac{6}{\delta^3}[\delta - \sinh(\delta)]e^x$, and $g(x) = e^x$. The same discretizations employed for the 1D problems with manufactured cubic and quadratic solutions are considered. Both first-order and second-order RK approximations with a cubic B-spline kernel and respective support sizes $a = 1.001h$ and $a = 2.1h$ are used. As described in Chapter 3, higher-order RK approximations can be considered by simply increasing the basis order in the basis vector $\mathbf{H}$. Figure 5.8 shows the convergence behavior obtained by using Gauss integration scheme 1 (results for Gauss integration scheme 2 can be found in [81]). A total of five Gauss integration points in each integration cell was employed in scheme 1 while a number of Gauss integration points equivalent to two Gauss integration points per cell was positioned in each horizon for scheme 2. For both cases, eight Gauss integration points per cell were used in the computation of the error L^2 norm. Similar results to the ones obtained for the problem with manufactured cubic solutions are obtained. For a linear RK approximation, a second-order convergence behavior is obtained, and rates higher than third-order are observed when a quadratic RK approximation is employed.

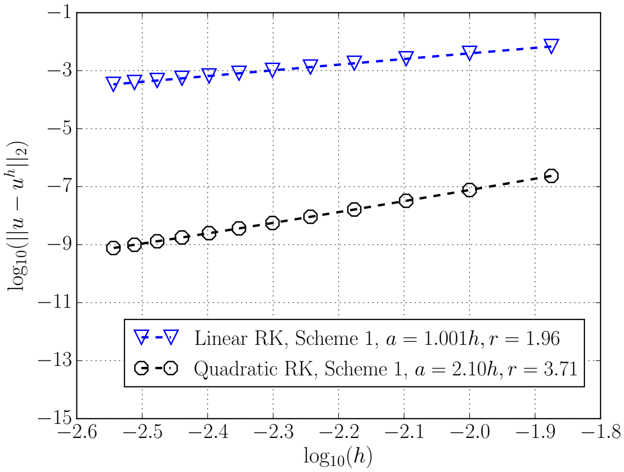


Figure 5.8: Convergence of the numerical solution of Eq. (5.36) for $u(x) = e^x$ using linear and quadratic RK approximations. Gauss integration scheme 1 for $\alpha = 0$ is used.

Study of the effect of the RK approximation support size

In the RK enhanced scheme, in addition to the horizon, which represents the support of the peridynamic influence function, the kernel support size a also needs to be assigned. Given that its value is linked only to the construction of the approximation function u^h, in this work a is chosen independently of the horizon δ. In this section, the effect of the choice of different RK support sizes a on the convergence behavior of the RK enhanced numerical solution is investigated. This is done by solving, for different values of a, the problems with manufactured quadratic, cubic, and exponential solutions presented in Section 5.3.1. Here the results for Gauss integration scheme 1 are presented; in [81] results related to Gauss integration scheme 2 for $\alpha = 0$ and $\alpha = 1$ can be found.

Figure 5.9 shows the obtained error results and the average convergence rates r for the 1D problem with manufactured quadratic solution using the Gauss integration scheme 1 for $\alpha = 0$ (see Figure 5.9). A linear RK approximation and the same data used in Section 5.3.1 were employed. A convergence rate close to second order is obtained regardless of the chosen RK support size.

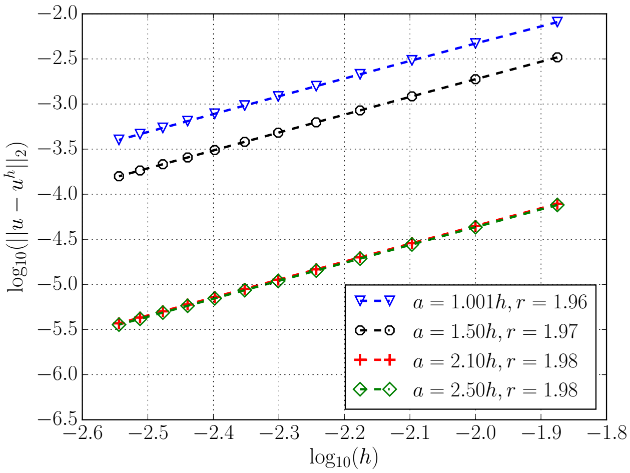
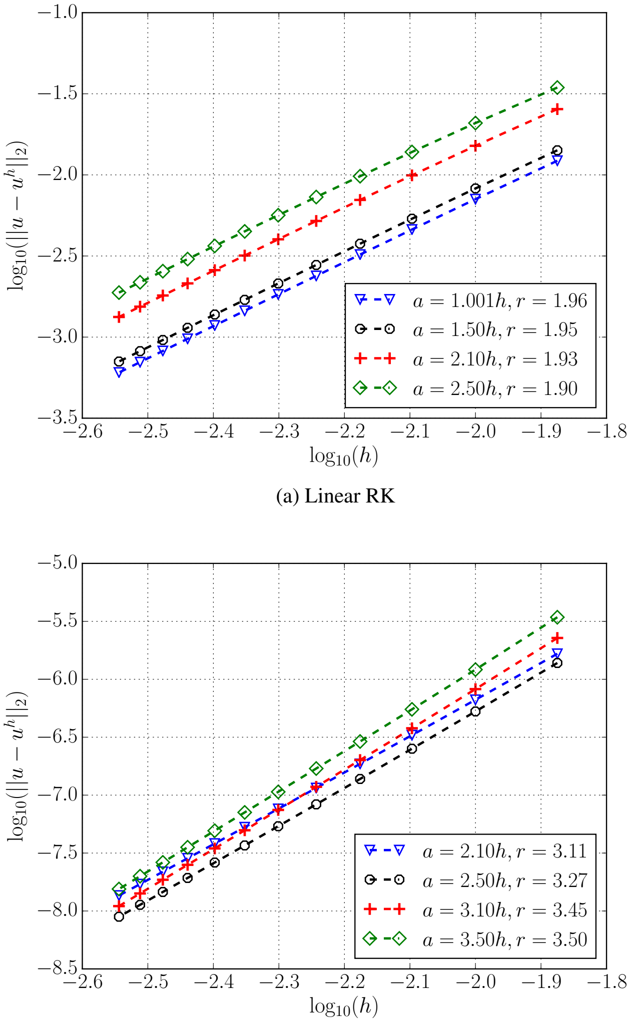


Figure 5.9: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$, $\alpha = 0$ in the influence function, and linear RK approximation using Gauss integration scheme 1 and different support sizes a.

The results obtained for different values of the RK support a for the 1D problem with manufactured cubic solution introduced in Section 5.3.1 are presented in Figures 5.10a and 5.10b for the case with $\alpha = 0$ in the influence function, which show the convergence results for the Gauss integration scheme 1 and linear and quadratic RK bases, respectively. For the linear basis, the convergence rates are close to second order for all the considered support sizes. For the quadratic basis, the convergence rate has a stronger variation with the support size but it is always higher than third order.

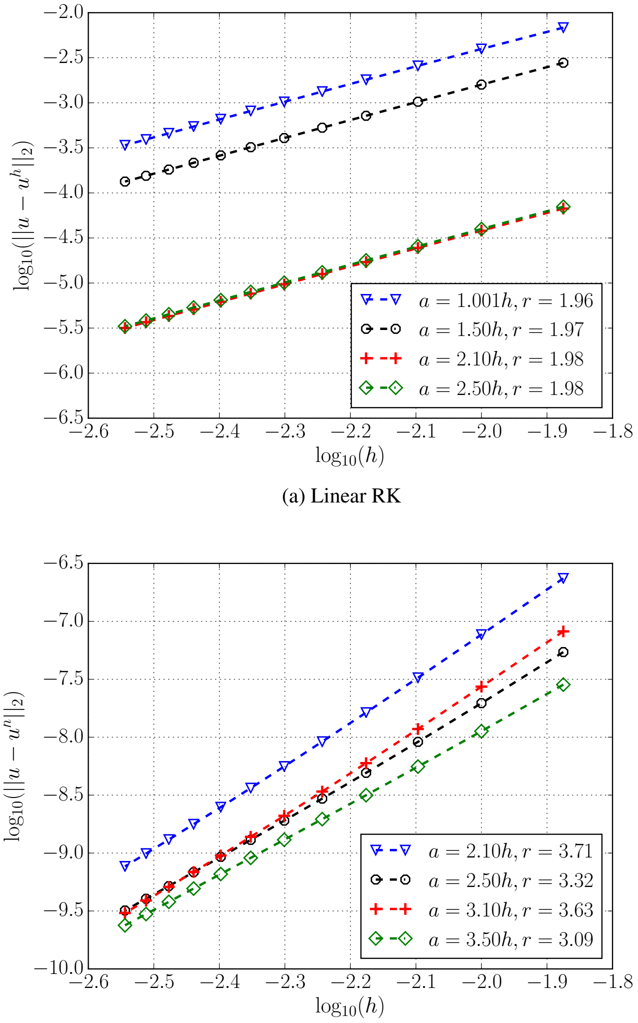


(a) Linear RK

(b) Quadratic RK

Figure 5.10: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^3$, $\alpha = 0$ in the influence function using Gauss integration scheme 1 and different support sizes a.

Lastly, the convergence behavior for different values of a in the solution of the problem with manufactured exponential solution introduced in Section 5.3.1 is presented. Figure 5.11a shows the obtained error results and the average convergence rates r when a linear RK approximation is used. The convergence rates are close to second order. The convergence behaviors obtained for a quadratic RK approximation are shown in Figure 5.11b. As for the case with manufactured cubic solutions, convergence rates close or higher than third order are obtained.



(a) Linear RK

(b) Quadratic RK

Figure 5.11: Convergence of the numerical solution of Eq. (5.36) for $u(x) = e^x$, $\alpha = 0$ in the influence function using Gauss integration scheme 1 and different support sizes a.

Study of the effect of the number of Gauss quadrature points

In this section we study the effect of the number of Gauss quadrature points on the convergence of the RK enhanced scheme. Integration error was identified as a major contributor to the L^2-error [93]. As shown in Appendix 5-A, if the integration in Eq. (5.36) is carried out using nodal integration, a highly oscillatory behavior is seen in the convergence results. As aforementioned in Section 5.3.1, in order to reduce the effect of integration error, high Gauss quadrature rules and the refinement scheme introduced in Section 5.3.1 were used. Here, one-node increment refinements are considered and the effect of the number of Gauss quadrature points on the convergence behavior is studied. For Gauss integration scheme 1, here considered (see [81] for the results for integration scheme 2), this is performed by increasing the number of Gauss points in each cell (NG_{cell}). The associated total number of Gauss points in the neighborhood ($NG1_\delta$) satisfies:

$$\left\lfloor \frac{\delta}{h} \right\rfloor \times NG_{cell} \leq NG1_\delta \leq \left\lceil \frac{\delta}{h} \right\rceil \times NG_{cell}.$$

The study of the effect of the number of Gauss quadrature points was carried out by solving the static linear problem with manufactured quadratic solution introduced in Section 5.3.1. The problem data is the same as in Section 5.3.1. A linear RK approximation with support size $a = 1.001h$ is employed. The convergence study is however performed by means of one-node increments starting from 76 nodes up to 151 nodes. An oscillatory behavior can be observed in Figure 5.12, where the convergence results are shown for different numbers of Gauss integration points per cell.

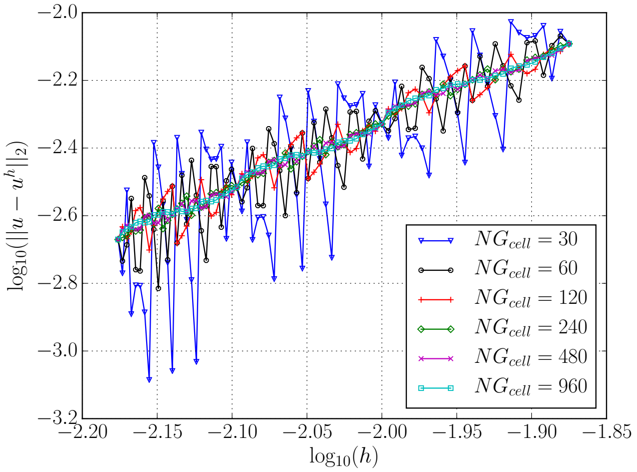


Figure 5.12: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear approximation for $\alpha = 0$ and different numbers of Gauss integration points per cell (NG_{cell}).

In Figure 5.12, it can be noted that as the number of Gauss integration points per cell increases, so does the integration accuracy, leading to a reduction in the oscillatory behavior. However, for all the number of Gauss integration points a periodic behavior can be observed in the convergence plots. This is likely related to the fact that the percentage of the two integration cells near the boundary of each nodal neighborhood that is included within the neighborhood varies periodically during the h-refinement. This type of behavior is therefore mitigated if the Gauss integration points are placed within the neighborhood of each node as in the Gauss integration scheme 2 (see [81]. When a high number of Gauss points per cell is used in Gauss integration scheme 1, the oscillatory behavior is strongly mitigated and an average convergence rate close to second order appears to be recovered. In fact, for $NG_{cell} = 960$, the average convergence rate is $r = 1.90$.

As noted by inspecting the results shown in Figure 5.12, in order to reduce the oscillatory behavior for Gauss integration scheme 1 a larger number of Gauss points is needed. In order to study how the increase in the number of Gauss points affects the accuracy of the solution for a given discretization, we solved the considered problem with quadratic manufactured solution for an increasing number of Gauss points. A discretization with 116 nodes was used. For such discretization the integration cells in Gauss integration scheme 1 are only partially covered by each nodal neighborhood. Figure 5.13 shows the error norm of the numerical solution for an increasing number of Gauss points in each integration cell. It can be observed that both schemes reach a similar accuracy as the number of Gauss points is increased. Gauss integration scheme 2 [81] though allows for the use of a lower number of integration points. For the Gauss integration scheme 2, the Gauss integration points are distributed inside each neighborhood as in Figure 5.2, so in this comparison, the total number of Gauss points in the neighborhood is chosen to be

$$NG2_{\delta} = \left\lceil \frac{\delta}{h} \right\rceil \times NG_{cell}, \tag{5.44}$$

where NG_{cell} is coincident with that from scheme 1 for the same nodal refinement, such that $NG2_{\delta}$ (total number of Gauss points in the neighborhood in scheme 2) and $NG1_{\delta}$ are comparable.

Overall, both schemes employ a larger number of integration points with respect to the node-based partial volume approach. However, the RK enhanced approach appears to yield better accuracy even for a relatively small amount of Gauss points, especially when Gauss integration scheme 2 is employed.

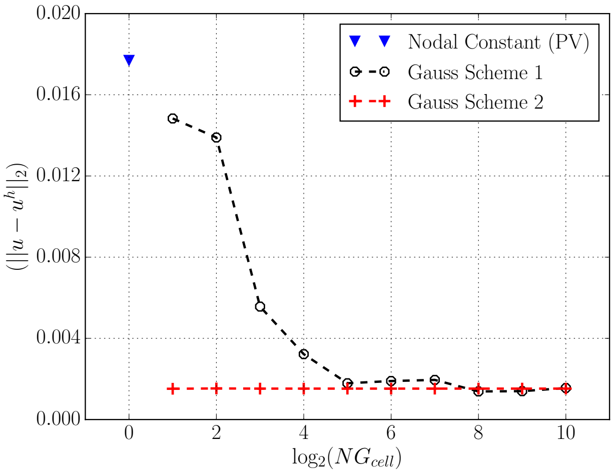


Figure 5.13: Error norm of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear RK approximation for $\alpha = 0$ for different number of Gauss points in each integration cells.

Non-uniform discretizations

Here, the convergence behavior of the proposed RK enhanced peridynamic scheme is investigated for non-uniform discretizations. We solve the static linear problem with manufactured quadratic solution introduced in Section 5.3.1, with $\alpha = 0$ in the influence function. The non-uniform discretizations were constructed by perturbing uniform discretizations with N nodes (and corresponding spacings of size h) in the problem domain interval $[0,1]$. This was achieved by moving each node in $(0,1)$ from their original position x_I in the uniform grid to a new randomly selected position $x_I^{nu} \in [x_I - \varepsilon h, x_I + \varepsilon h]$, where ε is a chosen perturbation factor. The problem is solved using an increasing number of nodes from 76 to 351 in increments of 25. The maximum spacing between two adjacent nodes in a non-uniform discretization is indicated as h_{max}. Figure

5.14 shows the results obtained using Gauss integration scheme 1 (for Gauss integration scheme 2 see [81]) for discretizations generated with $\varepsilon = 0.1, 0.2$. A total of 1024 Gauss points per cell was employed. Three Gauss points per cell were used in the computation of the error L^2 norm. The obtained convergence rates are all close to second order.

The need to use a high number of Gauss quadrature points to avoid oscillations in the convergence behavior arises from the same reasons presented in Section 5.3.1. In the convergence study with non-uniform discretizations, each refinement is composed of a random distribution of points. The nodal neighborhood is kept constant for all discretizations and, as such, it covers different number and portions of the integration cells. If the number of Gauss points is too low, the resulting integration error varies with each refinement, causing an oscillatory behavior similar to the one presented in Figure 5.12. This issue is mitigated when using Gauss integration scheme 2 [81].

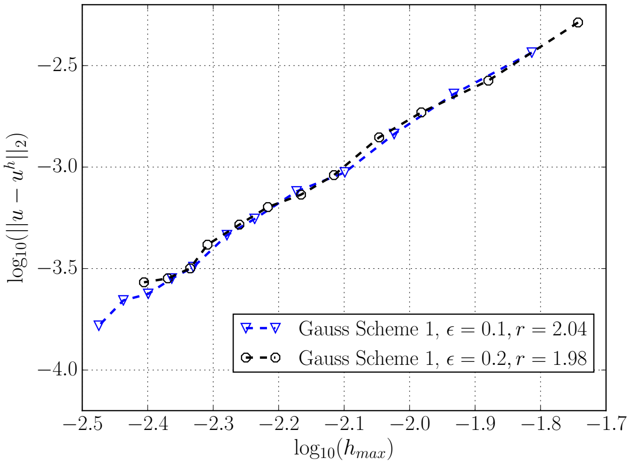


Figure 5.14: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using the linear approximation for $\alpha = 0$ and non-uniform discretizations arising from perturbation factors $\varepsilon = 0.1, 0.2$. Gauss integration schemes 1 is used.

5.3.2 Two-dimensional static peridynamic problem

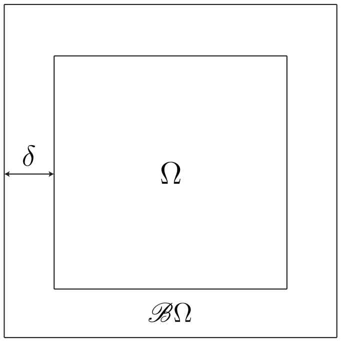


Figure 5.15: 2D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$.

Consider a unit square domain (Figure 5.15) $\overline{\overline{\Omega}} = [0,1] \times [0,1]$, with inner domain $\Omega = (\delta, 1-\delta) \times (\delta, 1-\delta)$ and boundary layer $\mathscr{B}\Omega = \overline{\overline{\Omega}} \setminus \Omega$, and a two-dimensional force state

$$\underline{\mathbf{T}}[\mathbf{x}]\langle\boldsymbol{\xi}\rangle = \frac{1}{2} c w(\|\boldsymbol{\xi}\|) \frac{\boldsymbol{\xi} \otimes \boldsymbol{\xi}}{\|\boldsymbol{\xi}\|^2} (\mathbf{u}(\mathbf{x}') - \mathbf{u}(\mathbf{x})), \tag{5.45}$$

where $\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x}$. The associated static linear bond-based peridynamic problem is

$$\begin{cases} -\int_{\mathscr{H}_{\mathbf{x}}} c w(\|\boldsymbol{\xi}\|) \frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|^2} (\mathbf{u}(\mathbf{x}') - \mathbf{u}(\mathbf{x})) dV_{\mathbf{x}'} = \mathbf{b}(\mathbf{x}), & \text{for } \mathbf{x} \in \Omega \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), & \text{for } \mathbf{x} \in \mathscr{B}\Omega \end{cases} \tag{5.46}$$

where $w(\|\boldsymbol{\xi}\|)$ is taken as

$$w(||\boldsymbol{\xi}||) = \begin{cases} \frac{1}{||\boldsymbol{\xi}||^\alpha}, & \text{for } ||\boldsymbol{\xi}|| \leq \delta \\ 0, & \text{for } ||\boldsymbol{\xi}|| > \delta \end{cases} \tag{5.47}$$

$\mathbf{g}(\mathbf{x})$ is a prescribed Dirichlet boundary condition function, and c is a constitutive constant taken from [93]:

$$c = \frac{6E}{m}, \tag{5.48}$$

where E is a constant and m is the weighted volume:

$$m = \int_{\mathscr{H}} w(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\|^2 d\boldsymbol{\xi} \tag{5.49}$$

with $\mathscr{H}$ representing a neighborhood of radius δ around the origin.

The procedure employed to construct the manufactured problem in two dimensions follows [93] and is similar to the one used for the one-dimensional case. A cubic polynomial solution form $\mathbf{u} = (u_1, u_2)$ is considered:

$$\begin{aligned} u_1(\mathbf{x}) &= m_1 x_1^3 + m_2 x_2^3 + m_3 x_1^2 x_2 + m_4 x_1 x_2^2 + m_5 x_1^2 \\ &\quad + m_6 x_2^2 + m_7 x_1 x_2 + m_8 x_1 + m_9 x_2 + m_{10}, \\ u_2(\mathbf{x}) &= n_1 x_1^3 + n_2 x_2^3 + n_3 x_1^2 x_2 + n_4 x_1 x_2^2 + n_5 x_1^2 \\ &\quad + n_6 x_2^2 + n_7 x_1 x_2 + n_8 x_1 + n_9 x_2 + n_{10}, \end{aligned} \tag{5.50}$$

where $m_1, ..., m_{10}$ and $n_1, ..., n_{10}$ are constants. The body force $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}))$ is obtained as

$$\begin{aligned} b_1(\mathbf{x}) &= -\frac{3E}{4}\left[(9m_1 + m_4 + 2n_3)x_1 + (3m_3 + 3m_2 + 2n_4)x_2 + 3m_5 + m_6 + n_7\right], \\ b_2(\mathbf{x}) &= -\frac{3E}{4}\left[(9n_2 + n_3 + 2m_4)x_2 + (3n_4 + 3n_1 + 2m_3)x_1 + 3n_6 + n_5 + m_7\right]. \end{aligned} \tag{5.51}$$

Convergence of numerical solutions in two dimensions

The convergence of the numerical solutions to the exact solution of the peridynamic problem in Eq. (5.46) is investigated. We first consider the case with a manufactured quadratic solution $\mathbf{u}(\mathbf{x}) = (x_1^2, 0)$. We choose $E = 1$, $\delta = 0.3$, $\mathbf{b}(\mathbf{x}) = (-9/4, 0)$, $\alpha = 0$, and $\mathbf{g}(\mathbf{x}) = (x_1^2, 0)$. The weighted volume m is computed analytically according to Eq. (5.49). Convergence is studied by solving the problem for an increasing number of uniformly spaced nodes. The same number of nodes is used in both directions (as in Figure 5.1); we start with 4 nodes ($\delta/h = 1.2$) in each direction (16 total) and increase that number until we reach a total number of 784 nodes by considering increasing ratios ($\delta/h = 1.2k$) with $k \in \mathbb{N}$. The choice of increasing the δ/h ratio by integer multiples of 1.2 was made to mimic in 2D the refinement performed in the quadratic 1D problem (Section 5.3.1). It has to be noted that different from the one-dimensional case, the refinement in the two-dimensional case does not allow for complete coverage of the integration cells by the neighborhood. Two numerical solution schemes are used: the node-based method discussed in Section 5.1.1 with full volumes and the RK enhanced approach with Gauss integration scheme 1 (Section 5.2). For the RK formulation a first-order basis, a cubic B-spline kernel with square support, and support sizes $a_1 = a_2 = 1.5h$ were used. Integration was performed by using 10x10 Gauss points in each integration cell. The same amount of Gauss integration points per cell was used to compute the error L^2 norm. Convergence results for the two methods are presented in Figure 5.16, where r is the average convergence rate fitting the whole data.

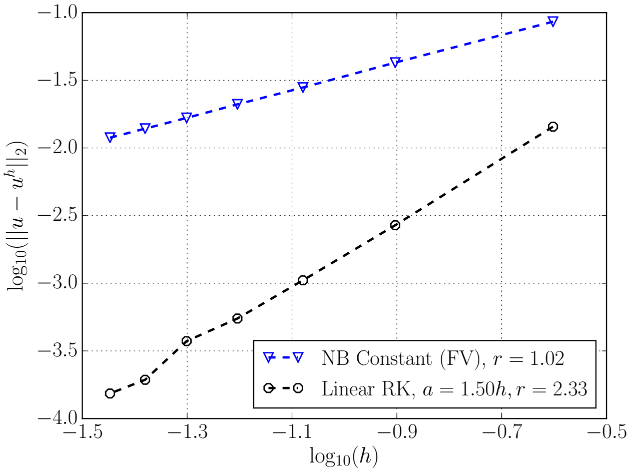


Figure 5.16: Convergence of the numerical solution of Eq. (5.46) for $\mathbf{u}(\mathbf{x}) = (x_1^2, 0)$, $\alpha = 0$ in Eq. (6.82), using the node-based meshfree solution with FV (NB Constant (FV)) and the linear RK approximation. Gauss integration scheme 1 is used.

The node-based method with full volumes exhibits a convergence rate close to first order. For the linear RK approximation the overall trend is closer to second order. A slightly oscillatory behavior, possibly attributable to integration error, is however observed.

5.3.3 Two-dimensional dynamic crack-branching problem

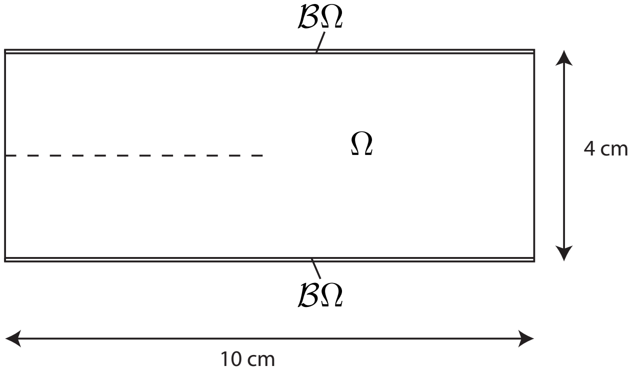


Figure 5.17: 2D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$. The pre-notch is indicated by the dashed line.

Given that peridynamics was originally developed from problems involving discontinuities (i.e., cracks and fractures), here we conduct a preliminary investigation of the performance of the RK Enhanced Peridynamic approach for a two-dimensional dynamic crack branching problem. Dynamic problems are described by Eq. (5.2) and differ from the static ones considered in the previous numerical examples mainly by the presence of the term $\rho(\mathbf{x})\frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x},t)$, which once the peridynamic equation is collocated at a node I, becomes $\rho(\mathbf{x_I})\frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x_I},t)$. The acceleration term, when the RK approximation is employed, can be recast in each direction $i = 1,...,d = 3$, as $\frac{\partial^2 u_i}{\partial t^2}(\mathbf{x_I},t) = \sum_{J=1}^{NP}\Psi_J(\mathbf{x}_I)\ddot{d}_{iJ}$.

Here, we consider a two-dimensional crack branching problem in soda-lime glass, first studied in [23]. The domain has dimensions 10 cm by 4 cm and the soda-lime glass has Young?s modulus $E = 72GPa$, density $\rho = 2440kg/m^3$, and energy release rate $G_0 = 3.8J/m^2$. We consider a constant Boolean unit influence function, a horizon $\delta = 2$ mm, and $m = \delta/h = 3$, where h is the uniform nodal spacing employed. This resulted in 160 and 60 nodes being used in the horizontal and vertical directions, respectively, for a total of N=9000 nodes. The material has been modeled using the Prototype Brittle Microelastic (PMB) material model ([112]). In the PMB material model the pairwise force density is:

$$\hat{\mathbf{f}}(\mathbf{x},\mathbf{x}',t) = \left(c \frac{\|\boldsymbol{\eta}+\boldsymbol{\xi}\| - \|\boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|} \right) \frac{\boldsymbol{\eta}+\boldsymbol{\xi}}{\|\boldsymbol{\eta}+\boldsymbol{\xi}\|} \tag{5.52}$$

where c is a constant, $\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x}$ and $\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}',t) - \mathbf{u}(\mathbf{x},t)$.

We consider a two-dimensional plane stress simulation, so we take [26]:

$$c = \frac{6E}{\pi\delta^3(1-\nu)} = \frac{9E}{\pi\delta^3}, \tag{5.53}$$

where the fact that for two-dimensional bond based peridynamics the Poisson's ratio $\nu = 1/3$ [132] has been used. For spatial integration, a node-based approach with Full Volume (FV) integration has been used. While it was shown in the previous numerical examples in this Section that such an approach does not always lead to sufficiently accurate integration for optimal convergence behavior, here it is used since it simply retains bond-breaking symmetry. If Gauss integration were used, in fact, bonds would have to be defined between each node $\mathbf{x}_I$ and each Gauss point $\mathbf{x}_{gJ}$ associated with every other discretization node $\mathbf{x}_J$. A horizontal pre-notch of length equal to 5 cm is created in the horizontal midline of the sample (see Figure 5.17): this is obtained by breaking all the bonds crossing the pre-notch and all the bonds associated with nodes located on the pre-notch itself. Over the course of the deformation, bonds break if their relative elongation is larger than the critical relative elongation, which is related to the choice of influence function and energy release rate. In this work:

$$s_0 = \sqrt{\frac{4\pi G_0}{9E}}. \tag{5.54}$$

The crack pattern is identified through the nodal damage variable φ, which is computed as

$$\varphi = 1 - \frac{\varphi_N}{\varphi_D}, \tag{5.55}$$

where φ_N is a nodal volume-weighted measure of the currently unbroken bonds, computed as the sum of the volumes associated with the neighboring nodes connected by unbroken bonds, while φ_D is the measure associated with the initial unbroken bonds, i.e., the original neighborhood volume. For time integration, the velocity-Verlet scheme is used with time step $\Delta t = 0.067\ \mu s$ and final simulation time $T = 43\ \mu s$.

First, we consider a tensile stress $\sigma = 2$ MPa, applied as a body force to the top and bottom rows of nodes (the boundary layer is taken to have width equal to the nodal spacing h). This will produce a single-branching crack [23]. The problem was solved by using the traditional node-based meshfree approach with full volumes and with a Lagrangian RK Enhanced approach with nodal integration, so that, as aforementioned, bonds and their breaking is defined between couple of nodes. Linear basis RK approximations with brick supports were employed. A normalized support size equal to 1.2 was chosen in both directions. A cubic B-spline kernel function was used. The result obtained with the node-based approach is shown in Figure 5.18. For the RK Enhanced approach, both a consistent and a lumped (via row-sum) density matrix were considered. The obtained results, which are shown Figures 5.19 and 5.20, respectively, match those obtained in [23] and qualitatively resemble the behaviour observed experimentally in [30] for cracking of soda-lime glass. It can be noted that for the considered cases, all obtained damage patterns are close to each other. This might be due to the fact that nodal integration can mitigate the effect of the introduction of the RK shape functions in the approximation (see Appendix 5-A).

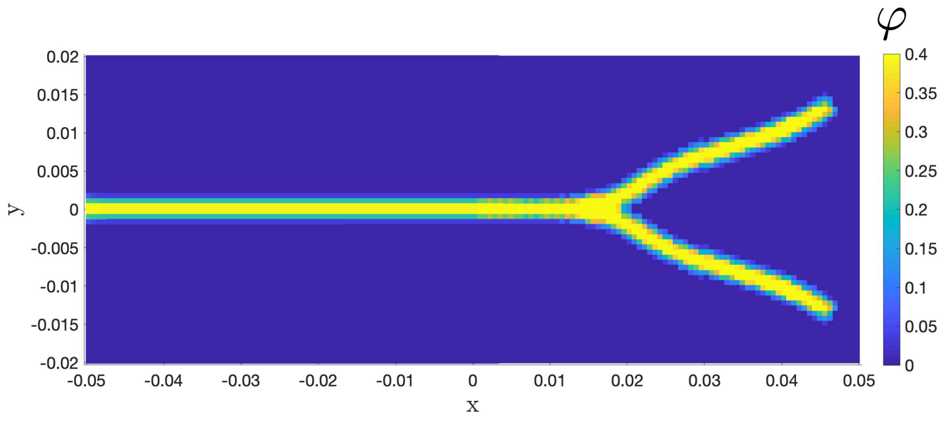


Figure 5.18: Fracture damage for the node-based meshfree solution.

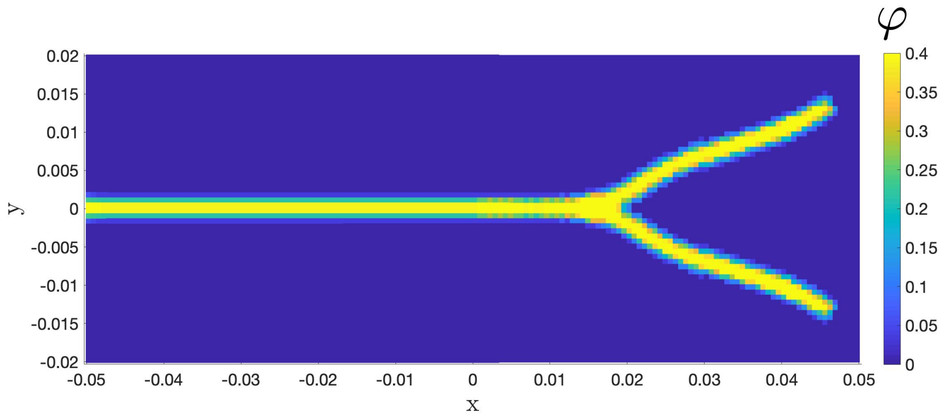


Figure 5.19: Fracture damage for the RK solution. A consistent density matrix was considered.

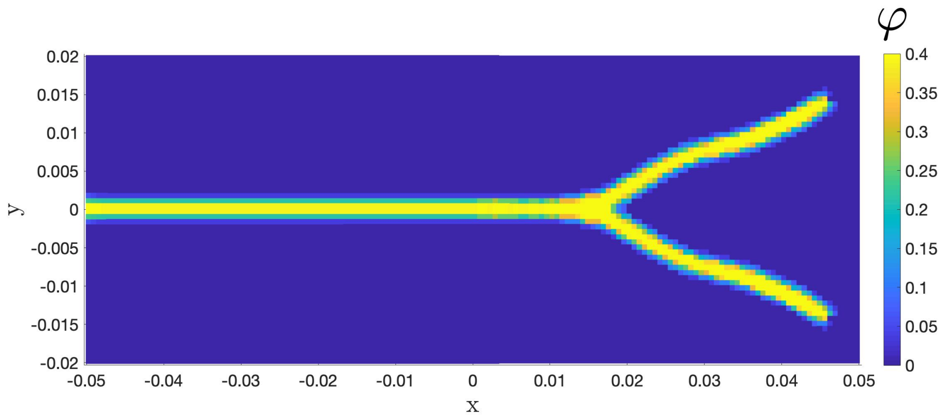


Figure 5.20: Fracture damage for the RK solution. A lumped density matrix was considered.

Next, a multiple-branching case was considered. The same parameters as in the single-branching case were considered with the exception of the applied tensile stress, which is now taken to be $\sigma = 4MPa$, and $m = \delta/h$ here taken equal to four. Again, the explicit velocity-Verlet scheme is used for time integration. The time step is taken to be $\Delta t = 0.05\mu s$ and the final simulation time is $T = 33\mu s$.The result obtained with the node-based approach is shown in Figure 5.21. For the RK Enhanced approach, both a consistent and a lumped (via row-sum) density matrix were considered. The obtained results are shown Figures 5.22 and 5.23 respectively. As for the single branching case, the observed overall behavior appears to be similar for the three considered cases. Again, these results match those obtained in [23] and qualitatively resemble the behaviour observed experimentally in [30].

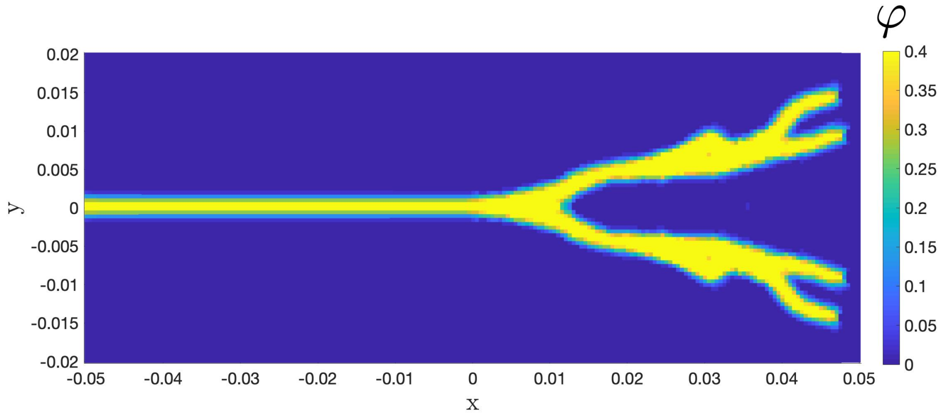


Figure 5.21: Fracture damage for the node-based meshfree solution.

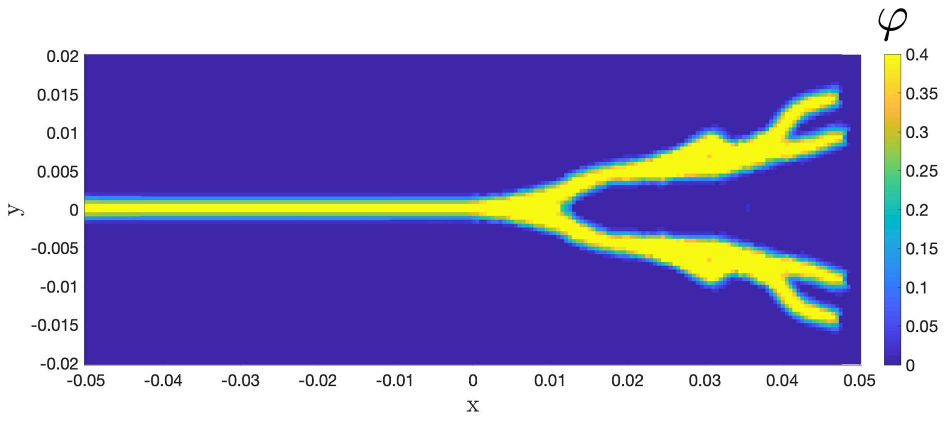


Figure 5.22: Fracture damage for the RK solution. A consistent density matrix was considered.

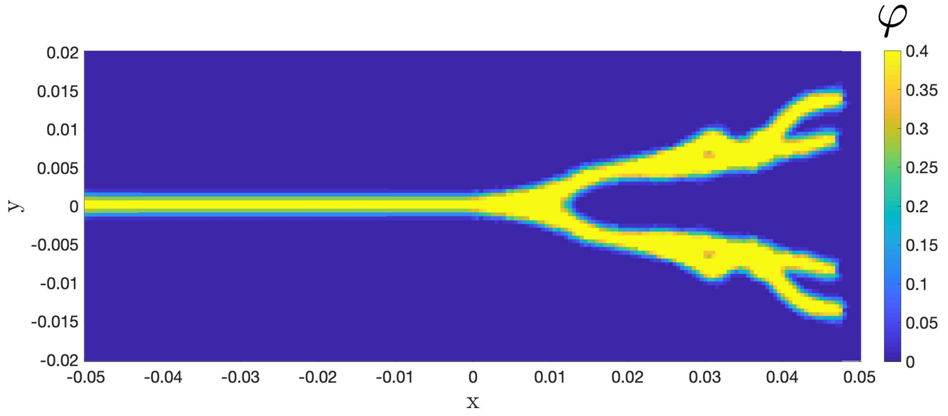


Figure 5.23: Fracture damage for the RK solution. A lumped density matrix was considered.

5.4 Peridynamic Modified Nodal Integration Weights

As highlighted in the previous Sections and in Appendix 5-A, accurate integration is paramount for achieving good accuracy and convergence behaviour of peridynamic solutions. This can be achieved, as described in Sections 5.2 and 5.3, through the use of a high-order Gauss quadrature scheme. However, high-order Gauss quadrature can become very expensive, especially in higher dimensions. On the other hand, a nodal integration quadrature scheme, while relatively computationally inexpensive does not lead sufficient integration accuracy (see

Appendix 5-A). Therefore, the main idea presented in this Section is to develop a high-order nodal integration quadrature. This is achieved by imposing $n-th$ order integration constraints (an idea presented for the weak form of classical mechanics in [61]) through reproducing conditions similar to those presented for RKPM in Chapter 3 ([79][78][80]). Let us consider an integration over the peridynamic neighbourhood of a node I of a generic function $f(\mathbf{x}') = f(\mathbf{x}_I, \mathbf{x}') = c\omega(\mathbf{x}_I, \mathbf{x})(u(\mathbf{x}') - u(\mathbf{x}_I))$ (see, for example, Eq. 5.36):

$$\int_{\mathscr{H}_{x_I}} f(\mathbf{x}')dV'_{\mathbf{x}}. \tag{5.56}$$

When integration over nodes $J = 1, ..., NP$ is used to approximate the term in Eq. (5.56), we get:

$$\int_{\mathscr{H}_{x_I}} f(\mathbf{x}')dV'_{\mathbf{x}} \approx \sum_{I \neq J, J=1}^{NP} f(\mathbf{x}_J)V_J. \tag{5.57}$$

where V_J is the integration weight associated with the J-th integration point.

Let us now consider for polynomials of order n, $u^n(\mathbf{x})$ (so that $f^n(\mathbf{x}') = c\omega(\mathbf{x}_I, \mathbf{x})(u^n(\mathbf{x}') - u^n(\mathbf{x}_I))$. Following a similar idea as in [61], we now want to impose exactness of the numerical nodal integration in Eq. (5.57) for polynomials of order n so that:

$$\int_{\mathscr{H}_{x_I}} f^n(\mathbf{x}')dV'_{\mathbf{x}} = \sum_{I \neq J, J=1}^{NP} f^n(\mathbf{x}_J)V_J. \tag{5.58}$$

Considering all orders of completeness $|k| = 1, ..., n$ we obtain:

$$\int_{\mathscr{H}_{x_I}} c\omega(\mathbf{x}_I, \mathbf{x})(\mathbf{x}'^k - \mathbf{x}_I^k)dV'_{\mathbf{x}} = \sum_{I \neq J, J=1}^{NP} (\mathbf{x}_J^k - \mathbf{x}_I^k)V_J, \;\; \forall 1 \leq |k| \leq n, \tag{5.59}$$

where the multi-index notation introduced in Chapter 3 has been used. Note that for a constant function u, Eq. (5.57) is trivially satisfied, so the first order of completeness we consider is $|k| = 1$.

Now, assuming the left-hand side in Eq. (5.59) is known and can be computed for a known influence function and order k, we use an idea similar to the one presented for RKPM in Chapter 3; that is, we want to construct integration weights so that the integral reproducing conditions in Eq. (5.59) are exactly satisfied. To this end, we replace each integration weight V_J with V_J^g, defined as a product of vectors of basis functions $\overline{\mathbf{H}}(\mathbf{x}_I, \mathbf{x}_J)$ and unknown coefficients $\overline{\mathbf{b}}$ to be determined [79][78][80]:

$$V_J^g = \overline{\mathbf{H}}_J^T \overline{\mathbf{b}}, \tag{5.60}$$

where $\overline{\mathbf{H}}_J^T = \left[\{\mathbf{x}_J^k - \mathbf{x}_I^k\}_{1 \leq |k| \leq n}\right]$. Substituting Eq. (5.60) into Eq. (5.59) we get:

$$\sum_{I \neq J, J=1}^{NP} (\mathbf{x}_J^k - \mathbf{x}_I^k) \overline{\mathbf{H}}_J^T \overline{\mathbf{b}} = \beta_k \; \forall 1 \leq |k| \leq n, \tag{5.61}$$

where we defined

$$\beta_k = \int_{\mathscr{H}_{x_I}} c\omega(\mathbf{x}_I, \mathbf{x})(\mathbf{x}'^k - \mathbf{x}_I^k) dV'_{\mathbf{x}}. \tag{5.62}$$

Now, Eq. (5.67) leads to the following system:

$$\overline{\mathbf{M}\mathbf{b}} = \boldsymbol{\beta}, \tag{5.63}$$

where $\boldsymbol{\beta}$ is a column vector containing the k exact integrals in Eq.(5.62) and

$$\overline{\mathbf{M}} = \sum_{I \neq J, J=1}^{NP} \overline{\mathbf{H}}_J \overline{\mathbf{H}}_J^T. \tag{5.64}$$

It can be noted that $\overline{\mathbf{M}}$ is similar to the moment matrix obtained in the RKPM construction (see Chapter 3). If $\overline{\mathbf{M}}$ is invertible, from Eq. (5.63) we get:

$$\overline{\mathbf{b}} = \overline{\mathbf{M}}^{-1} \boldsymbol{\beta}, \tag{5.65}$$

which substituted in Eq. (5.60) leads to the expression for the integration weights:

$$V_J^g = \overline{\mathbf{H}}_J^T \overline{\mathbf{M}}^{-1} \boldsymbol{\beta}. \tag{5.66}$$

Now, invertibility of $\overline{\mathbf{M}}$ is not guaranteed and can be linked to the non-uniqueness of the integration weights satisfying the constraints in Eq. (5.67). As an example, we can consider a one-dimensional case with a uniform distribution of nodes. For the first-order condition (i.e., $k = 1$) $\beta_k = 0$ we can get from Eq. (5.67):

$$\sum_{I \neq J, J=1}^{NP} (x_J - x_I)\overline{\mathbf{H}}_J^T \overline{\mathbf{b}} = \sum_{I \neq J, J=1}^{NP} (x_J - x_I) V_J^g = 0, \tag{5.67}$$

which is satisfied either by zero-weights $V_J^g = 0$, $\forall J$ or by constant symmetrical weights $V_J^g = V$, $\forall J$. In Appendix 5-B we show for a one-dimensional case with a uniform discretization that for symmetrical integration weights the equations associated with odd k are automatically satisfied (i.e., lead to $0 = 0$). If included, they would produce zero rows in $\overline{\mathbf{M}}$. This can be circumvented by only imposing the even k conditions.

For illustration, let us now consider the one-dimensional example presented in Section 5.3.1 for a quadratic manufactured solution. Figures 5.24 and 5.25, show the obtained displacement fields by using the Full Volume (FV) integration approach and nodal integration with the weights derived in Eq. (5.66) after imposing second-order consistency. It can be seen that the modified weights allow recovering of the exact manufactured solution.

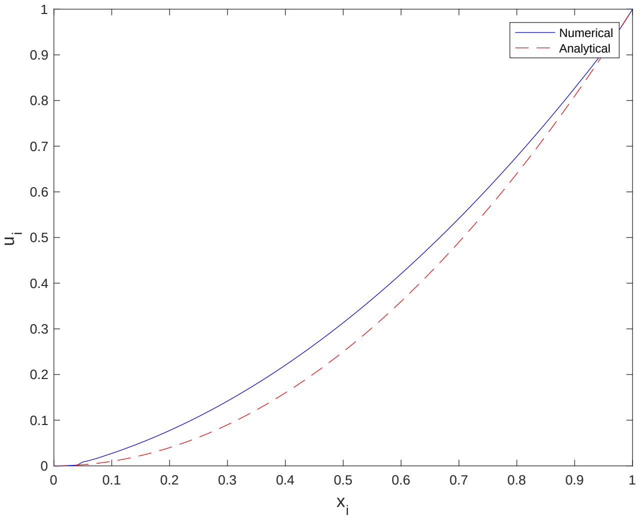


Figure 5.24: Solution for one-dimensional peridynamic problem with manufactured solution using a FV integration approach.

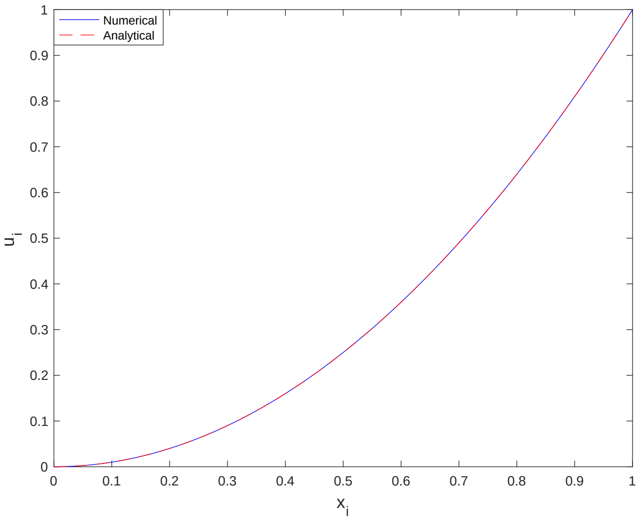


Figure 5.25: Solution for one-dimensional peridynamic problem with manufactured solution using nodal integration with modified integration weights with quadratic consistency.

Lastly, it can be shown that this approach of deriving integration weights is equivalent to the one presented in [88], when the same functions are to be integrated (i.e., $f(\mathbf{x}') = c\omega(\mathbf{x}_I,\mathbf{x})(u(\mathbf{x}') - u(\mathbf{x}_I)))$ and the same basis vector $\overline{\mathbf{H}}$ is used. In [88] the quadrature weights are found through the Generalized Moving Least Square (GMLS) [104] by imposing integration exactness of functions through a least squares approach. The resulting expression for the quadrature weights is [88]:

$$\boldsymbol{\omega} = \mathbf{B}^T\mathbf{S}^{-1}\mathbf{g}, \tag{5.68}$$

where $\boldsymbol{\omega}^T = \{\omega_1, ..., \omega_J, ..., \omega_{NP}\}$, $\forall J \neq I$, is the vector of integration weights of size equal to the number of quadrature points in the family of node I. $\mathbf{g}$ is the vector containing the k exact integral g_k

$$g_k = \int_{\mathscr{H}_{x_I}} c\omega(\mathbf{x}_I,\mathbf{x})(\mathbf{x}'^k - \mathbf{x}_I^k)dV'_{\mathbf{x}}, \tag{5.69}$$

and $\mathbf{B}$ is a rectangular matrix with column vectors $\overline{\mathbf{H}}_J$, $\forall J \neq I$, and $\mathbf{S} = \mathbf{B}\mathbf{B}^T$. It follows that $\mathbf{g} = \boldsymbol{\beta}$, $\mathbf{S} = \overline{\mathbf{M}}$ and that each integration weight ω_J in $\boldsymbol{\omega}$ is:

$$\omega_J = \mathbf{H}_J^T\mathbf{S}^{-1}\mathbf{g} = \mathbf{H}_J^T\overline{\mathbf{M}}^{-1}\boldsymbol{\beta}. \tag{5.70}$$

By comparing Eq. (5.66) and Eq. (5.70) it is clear that the weights computed with the two approaches are the same. This is not too surprising given the fact that both approaches are imposing reproduction of integral of functions, though one directly and one through a minimization problem. This relates to the similarity between RKPM and RK implicit gradient [66] and GMLS, which imposes reproductions of generic functionals through a minimization approach.

Acknowledgments

A portion of this chapter has been published in M. Pasetto, Y. Leng, J.S. Chen, J.T. Foster, P. Seleson, “A reproducing kernel enhanced approach for peridynamic solutions”, *Computer Methods in Applied Mechanics and Engineering*, 340, 2018. The dissertation author was the

primary investigator of the included material.

A portion of this chapter is currently being prepared for submission for publication of the material. M. Pasetto, J.S. Chen, “Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics”. The dissertation author was the primary investigator of this material.

5.5 Appendix 5-A: Effect of nodal integration on convergence

Here, it is shown that the error associated with integration strongly affects the convergence behavior, even when the RK enhanced approach is used. To do so, the one-dimensional peridynamic problem with manufactured quadratic solution $u(x) = x^2$ introduced in Section 5.3.1 is solved by using a nodal integration approach. Both the node-based method discussed in Section 5.3.1 with partial volumes and the RK enhanced approach are considered. The same data employed in Section 5.3.1 are used: $\alpha = 0$, $K = 1$, $\delta = 0.04$, $b(x) = -2$, and $g(x) = x^2$. The weighted volume m is computed analytically according to Eq. (6.84). Convergence is studied by solving the problem for an increasing number of uniformly spaced nodes: we start with 76 nodes ($\delta/h = 3$) and increase the number of nodes by 1 until we reach 151 nodes ($\delta/h = 6$). For the RK formulation used in this example problem, a first-order basis and a second-order basis were used. A support size of $a = 1.001h$ was used for the case with linear bases, while $a = 2.1h$ was chosen for the case with quadratic bases. The convergence behavior of a node-based solution described by Eq. (5.36) was studied for this problem in [93] for different influence functions and integration weights. Figure 5.26 shows the error convergence results for the RK enhanced approach with nodal integration, and compares them with the error obtained in [93] for a piecewise-constant approximation where the PV approach (NB Constant (PV)) described in Section 5.3.1 was used.

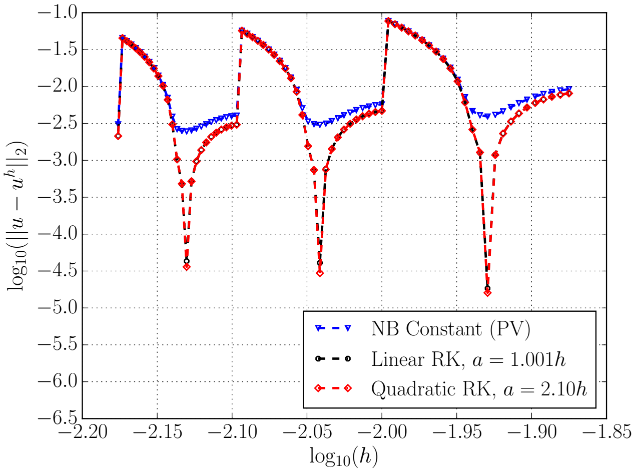


Figure 5.26: Convergence of the numerical solution of Eq. (5.36) for $u(x) = x^2$ using node-based piecewise-constant meshfree solution (with PV) and RK approximations. Integration error strongly affects the convergence behavior.

It can be observed that, even though the completeness of the approximation was increased through the use of the RK approximation, integration error significantly affects the convergence behavior. Reducing the effect of the integration error is thus necessary to achieve optimal convergence behaviors.

5.6 Appendix 5-B: Reproducing conditions for integration weights in one-dimensional uniform discretizations

Let us start by considering an arbitrary polynomial of order $2z$:

$$u(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n + \ldots a_{2z} x^{2z} = \sum_{n=0}^{2z} a_n x^n \tag{5.71}$$

Let us consider

$$\int_{x-\delta}^{x+\delta} c\omega[u(x') - u(x)]dx' = \int_{-\delta}^{+\delta} c\omega[u(x+\xi) - u(x)]d\xi, \tag{5.72}$$

where ω is used to denote $\omega(|x'-x|)$, $\omega(|\xi|)$. Eq. (5.72) becomes

$$\int_{-\delta}^{+\delta} c\omega[u(x+\xi)-u(x)]d\xi = \sum_{n=1}^{2z}\int_{-\delta}^{+\delta} c\omega[a_n(x+\xi)^n - a_n x^n]d\xi \tag{5.73}$$

By using the binomial theorem:

$$\begin{aligned}\sum_{n=1}^{2z}\int_{-\delta}^{+\delta} c\omega\{a_n[(x+\xi)^n - x^n]\}d\xi &= \sum_{n=1}^{2z}\int_{-\delta}^{+\delta} c\omega\left\{a_n\left[\sum_{p=0}^{n}\binom{n}{p}x^{n-p}\xi^p - x^n\right]\right\}d\xi \\ &= \sum_{n=1}^{2z}\int_{-\delta}^{+\delta} c\omega\left\{a_n\left[\sum_{p=1}^{n}\binom{n}{p}x^{n-p}\xi^p\right]\right\}d\xi \\ &= \sum_{n=1}^{2z} a_n \sum_{p=1}^{n}\binom{n}{p}x^{n-p}\int_{-\delta}^{+\delta} c\omega\xi^p d\xi\end{aligned} \tag{5.74}$$

Now, let us consider the right-hand side of Eq. (5.74):

$$\begin{aligned}& a_n\sum_{p=1}^{n}\binom{n}{p}x^{n-p}\int_{-\delta}^{+\delta} c\omega\xi^p d\xi \\ &= a_n\left[\sum_{p=1}^{n}\binom{n}{p}x^{n-p}\int_{-\delta}^{+\delta} c\omega\xi^p d\xi\right] \\ &= a_n\left[\sum_{p=1}^{n}\frac{n!}{p!(n-p)!}x^{n-p}\int_{-\delta}^{+\delta} c\omega\xi^p d\xi\right] \\ &= a_n\left[\sum_{p=1}^{n}\frac{n!}{(n-p)!}x^{n-p}m_p\right] \\ &= a_n[\beta_n]\end{aligned} \tag{5.75}$$

where

$$m_p = \frac{1}{p!}\int_{-\delta}^{\delta} c\omega\xi^p d\xi. \tag{5.76}$$

Eq. (5.74) then becomes:

$$\sum_{n=1}^{2z}\int_{-\delta}^{+\delta} c\omega\{a_n[(x+\xi)^n - x^n]\}d\xi = \sum_{n=1}^{2z} a_n\left[\sum_{p=1}^{n}\frac{n!}{(n-p)!}x^{n-p}m_p\right] \tag{5.77}$$

In order to find the integration weights, we consider Eq. (5.77) being collocated at $x = x_I$ as well as a nodal discretization of the peridynamic integral. The nodal discretization leads in general to a summation over all the nodes x_J in the family $\mathcal{F}_I$ of x_I. In order to be able to compare the two considered approaches we consider a uniformly discretized case with the same number of neighbours: hence there are z neighbours to the left of x_I and z to the right of x_I.

$$\sum_{n=1}^{2z} a_n \sum_{J\in\mathcal{F}_I} c\omega_{IJ}(x_J^n - x_I^n)V_J^g = \sum_{n=1}^{2z} a_n \left[\sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p\right] \tag{5.78}$$

$$\sum_{n=1}^{2z} a_n \sum_{k=-z}^{z} c\omega_k \left[(x_I + kh)^n - x_I^n\right] V_k^g = \sum_{n=1}^{2z} a_n \left[\sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p\right] \tag{5.79}$$

$$\sum_{n=1}^{2z} a_n \sum_{k=-z}^{z} c\omega_k \left[\sum_{p=0}^{n} \binom{n}{p} (x_I)^{n-p}(kh)^p - x_I^n\right] V_k^g = \sum_{n=1}^{2z} a_n \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.80}$$

$$\sum_{n=1}^{2z} a_n \sum_{k=-z}^{z} c\omega_k \sum_{p=1}^{n} \binom{n}{p} (x_I)^{n-p}(kh)^p V_k^g = \sum_{n=1}^{2z} a_n \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.81}$$

Now, in this approach, in order to find V_k^g, we impose Eq. (5.81) $\forall n$, $n = 1, ..., 2z$:

$$\sum_{k=-z}^{z} c\omega_k \sum_{p=1}^{n} \binom{n}{p} (x_I)^{n-p}(kh)^p V_k^g = \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.82}$$

$$\sum_{k=-z}^{z} c\omega_k \sum_{p=1}^{n} \frac{n!}{p!(n-p)!} (x_I)^{n-p}(kh)^p V_k^g = \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.83}$$

$$\sum_{p=1}^{n} \frac{n!}{p!(n-p)!} (x_I)^{n-p} \sum_{k=-z}^{z} c\omega_k (kh)^p V_k^g = \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.84}$$

$$\sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} \sum_{k=-z}^{z} \frac{1}{p!} c\omega_k (kh)^p V_k^g = \sum_{p=1}^{n} \frac{n!}{(n-p)!} x_I^{n-p} m_p \tag{5.85}$$

By considering each p separately (see **Remark 1**):

$$\frac{1}{p!} \sum_{k=-z}^{z} c\omega_k (kh)^p V_k^g = m_p, \ \ \forall p = 1, ..., n \tag{5.86}$$

which, if $n = 2z$, is:

$$\frac{1}{p!}\sum_{k=-z}^{z} c\omega_k (kh)^p V_k^g = m_p, \;\; \forall p = 1, ..., 2z \tag{5.87}$$

Now, for odd values of p:

$$m_p = \frac{1}{p!}\int_{-\delta}^{\delta} c\omega\xi^p d\xi = 0 \tag{5.88}$$

$$\frac{1}{p!}\sum_{k=-z}^{z} c\omega_k (kh)^p V_k^g = 0 \tag{5.89}$$

Eq. (5.87) is satisfied $\forall p$ only either if all the weights are zero (non admissible due to the conditions imposed when p is even) or if the integration weights are symmetrical, i.e. $V_{-k} = V_k$. Symmetrical weights satisfy the conditions imposed for odd values of p; hence, we are left with the conditions for the even values of p:

$$\frac{1}{p!}\sum_{k=-z}^{z} c\omega_k (kh)^p V_k^g = m_p, \;\; \forall p \in 2\mathbb{N}, p = 2, ..., 2z \tag{5.90}$$

$$\frac{1}{p!}\sum_{k=1}^{z} 2c\omega_k (kh)^p V_k^g = \frac{1}{p!}\int_{-\delta}^{\delta} c\omega\xi^p d\xi, \;\; \forall p \in 2\mathbb{N}, p = 2, ..., 2z \tag{5.91}$$

Remark 1: the reason that allows us to consider separately each p going from Eq. (5.85) to Eq. (5.86) is rooted in the fact that $n = 1, ..., 2z$ and that we are solving one equation for each value of n. So when $n = 1$ in Eq. (5.85), $p = 1$ and we have the equation for $p = 1$, then when $n = 2$, $p = 1, 2$; the condition for $p = 1$ is already satisfied to the imposition of the same condition when $n = 1$ and so on. This same reasoning applies for all $n = 1, ..., 2z$. See the following system for clarification:

$$\begin{cases} \sum_{k=-z}^{z} c\omega_k (kh) V_k^g = m_1 \\ 2x_I \sum_{k=-z}^{z} c\omega_k (kh) V_k^g + 2! \sum_{k=-z}^{z} c\omega_k \frac{1}{2!} (kh)^2 V_k^g = 2x_I m_1 + 2! m_2 \\ \frac{3!}{2!} x_I^2 \sum_{k=-z}^{z} c\omega_k (kh) V_k^g + 3! x_I \sum_{k=-z}^{z} c\omega_k \frac{1}{2!} (kh)^2 V_k^g \\ +3! \sum_{k=-z}^{z} c\omega_k \frac{1}{3!} (kh)^3 V_k^g = \frac{3!}{2!} x_I^2 m_1 + 3! x_I m_2 + 3! m_3 \\ \frac{4!}{3!} x_I^3 \sum_{k=-z}^{z} c\omega_k (kh) V_k^g + \frac{4!}{2!} x_I^2 \sum_{k=-z}^{z} c\omega_k \frac{1}{2!} (kh)^2 V_k^g \\ +4! x_I \sum_{k=-z}^{z} c\omega_k \frac{1}{3!} (kh)^3 V_k^g + 4! \sum_{k=-z}^{z} c\omega_k \frac{1}{4!} (kh)^4 V_k^g \\ = \frac{4!}{3!} x_I^3 m_1 + \frac{4!}{2!} x_I^2 m_2 + 4! x_I m_3 + 4! m_4 \\ \ldots \\ \sum_{p=1}^{2z} \frac{2z!}{(2z-p)!} x_I^{2z-p} \sum_{k=-z}^{z} c\omega_k \frac{1}{p!} (kh)^p V_k^g = \sum_{p=1}^{2z} \frac{2z!}{(2z-p)!} x_I^{2z-p} m_p \end{cases} \tag{5.92}$$

that can be rearranged as

$$\begin{cases} \sum_{k=-z}^{z} c\omega_k (kh) V_k^g = m_1 \\ \sum_{k=-z}^{z} c\omega_k \frac{1}{2!} (kh)^2 V_k^g = m_2 \\ \sum_{k=-z}^{z} c\omega_k \frac{1}{3!} (kh)^3 V_k^g = m_3 \\ \sum_{k=-z}^{z} c\omega_k \frac{1}{4!} (kh)^4 V_k^g = m_4 \\ \ldots \\ \sum_{k=-z}^{z} c\omega_k \frac{1}{2z!} (kh)^{2z} V_k^g = m_{2z} \end{cases} \tag{5.93}$$

Chapter 6

Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics

As for classical PDE-based boundary/initial value problems, the peridynamic strong form formulation can be recast into a so-called variational (or weak) form, which mimics the one typically employed for the solution of partial differential equations. Since the variational form of the peridynamic balance of linear momentum entails the solution of a double integral, it generally involves significant geometric complexity and requires higher computational expense relative to the strong form [12]. For this reason, even though the use of the weak form has been recently proposed to facilitate the imposition of nonlocal boundary conditions [22], it has been mainly employed for the purpose of proving theorems about its properties, convergence rates for different approximations, and the uniqueness and existence of a solution [69][141][142][125][99], rather than for practical applications, for which it has been used mainly in conjunction with finite element approximations [140][69]. Nonetheless, due to the presence of the double integration, the discrete peridynamic weak form can retain symmetry of interaction between two points in the domain, which is often lacking in discrete strong peridynamic formulations [8]. Here, we propose a variationally consistent integration for weak form peridynamics, which allows for exact integration for polynomial solutions up to a desired order, while retaining the symmetrical feature of weak form peridynamics.

6.1 Review of nonlocal vector calculus

Nonlocal vector calculus was originated from the study of peridynamics [72][102] and provides a systematic way of formulating nonlocal models, such as peridynamics [69], through the use of nonlocal operators. In this Section we review some of its basic aspects and provide the definitions of the nonlocal divergence and gradient operators. Furthermore, some of the theorems of nonlocal vector calculus, such as nonlocal integration by parts, will be reviewed so to introduce the weak form of the peridynamic equation of motion. The interested reader is referred to [72][102][69][103] and [5] for further discussions on nonlocal vector calculus.

To start, consider $\mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{R}^n$. For any point $\mathbf{x} \in \mathbb{R}^n$, and for integrable functions $\psi(\mathbf{x},\mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, we define [102]

$$\int_{\widetilde{\Omega}} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y} \;\; \forall\, \widetilde{\Omega} \subseteq \mathbb{R}^n, \tag{6.1}$$

as the nonlocal flux density at $\mathbf{x}$ into a measurable volume $\widetilde{\Omega}$. Therefore, for any two open measurable regions $\Omega_1 \subseteq \mathbb{R}^n$ and $\Omega_2 \subseteq \mathbb{R}^n$

$$\int_{\Omega_1} \int_{\Omega_2} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y} d\mathbf{x} \tag{6.2}$$

represents a scalar interaction (nonlocal flux) from Ω_1 into Ω_2. In accordance with Eq. (6.1), the term $\int_{\Omega_2} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y}$ in Eq. (6.2) is the nonlocal flux density at $\mathbf{x} \in \Omega_1$ into the region Ω_2. Now, we require no self-interaction, i.e. the flux from a region into itself vanishes [102][5]:

$$\int_{\widetilde{\Omega}} \int_{\widetilde{\Omega}} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y} d\mathbf{x} = 0, \;\; \forall\, \widetilde{\Omega} \subseteq \mathbb{R}^n, \tag{6.3}$$

which is equivalent to requiring that $\psi(\mathbf{x},\mathbf{y}) = -\psi(\mathbf{y},\mathbf{x})$ for almost all $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$. Eq (6.3) is obviously satisfied if we consider an antisymmetric function [5]

$$\psi(\mathbf{x},\mathbf{y}) = -\psi(\mathbf{y},\mathbf{x}), \;\; \forall\, \mathbf{x},\mathbf{y} \in \mathbb{R}^n. \tag{6.4}$$

Futhermore, it is proved in [102] that Eq. (6.3) is equivalent to

$$\int_{\Omega_1} \int_{\Omega_2} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y} d\mathbf{x} + \int_{\Omega_2} \int_{\Omega_1} \psi(\mathbf{x},\mathbf{y}) d\mathbf{y} d\mathbf{x} = 0, \;\; \forall\, \Omega_1, \Omega_2 \subseteq \mathbb{R}^n, \tag{6.5}$$

that is the nonlocal flux from Ω_1 into Ω_2 is equal and opposite to the nonlocal flux from Ω_2 into Ω_1. Eq. (6.5) is therefore the nonlocal analogue of the local action-reaction principle [102].

Now, a nonlocal operator $\mathcal{D}$ is sought so that it satisfies a nonlocal analogue of the divergence theorem, i.e., "the integral of the nonlocal divergence of a vector $\boldsymbol{\nu}$ over any region $\widetilde{\Omega} \subset \mathbb{R}^n$ is equal to the flux out of that region" [102]:

$$\int_{\widetilde{\Omega}} \mathcal{D}\boldsymbol{\nu}(\mathbf{x})d\mathbf{x} = \int_{\widetilde{\Omega}} \int_{\mathbb{R}^n} \psi(\mathbf{x},\mathbf{y})d\mathbf{y}d\mathbf{x}, \ \forall\, \widetilde{\Omega} \subseteq \mathbb{R}^n, \tag{6.6}$$

where we have considered that a point $\mathbf{x} \in \widetilde{\Omega}$ can interact with all points $\mathbf{y} \in \mathbb{R}^n$ so that the nonlocal flux density out of $\mathbf{x}$ is [102]

$$\int_{\mathbb{R}^n} \psi(\mathbf{x},\mathbf{y})d\mathbf{y}. \tag{6.7}$$

The operator $\mathcal{D}$ is then called nonlocal divergence.

In [5][102] a linear and continuous map $\mathcal{D} : [C_c^\infty(\mathbb{R}^n \times \mathbb{R}^n)]^k \to D'(\mathbb{R}^n)$, with $k = 1, 2, ..., 3$ and where $D'(\mathbb{R}^n)$ (space of distributions) is the dual space of $D = C_c^\infty(\mathbb{R}^n)$, is first considered. Then, for almost all $\mathbf{x} \in \mathbb{R}^n$, from the Schwartz kernel theorem,

$$\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) = \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}, \tag{6.8}$$

where the kernel $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \in ([C_c^\infty(\mathbb{R}^n \times \mathbb{R}^n)]^k \times C_c^\infty(\mathbb{R}^n))'$. Now, by looking at Eq. (6.6)

$$\int_{\widetilde{\Omega}} \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}d\mathbf{x} = \int_{\widetilde{\Omega}} \int_{\mathbb{R}^n} \psi(\mathbf{x},\mathbf{y})d\mathbf{y}d\mathbf{x}, \ \forall\, \widetilde{\Omega} \subseteq \mathbb{R}^n, \tag{6.9}$$

the nonlocal flux is expressed in [102] as

$$\psi(\mathbf{x},\mathbf{y}) = \int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}, \ \textit{for almost all} \ \mathbf{x},\mathbf{y} \in \mathbb{R}^n, \tag{6.10}$$

which, by the antisymmetry of $\psi(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{x}$ and $\mathbf{y}$ (see Eq. (6.4)), requires [102]

$$\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) = -\boldsymbol{\kappa}(\mathbf{y},\mathbf{x},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{x}), \ \textit{for almost all} \ \mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{R}^n. \tag{6.11}$$

Therefore, for a fixed kernel $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z})$, the flux density per unit volume $\psi(\mathbf{x},\mathbf{y})$ is uniquely expressed in terms of $\boldsymbol{\nu}(\mathbf{x},\mathbf{y})$ in [102] as

$$\psi(\mathbf{x},\mathbf{y}) = \frac{1}{2}\int_{\mathbb{R}^n} \left[\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y}) - \boldsymbol{\kappa}(\mathbf{y},\mathbf{x},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{x})\right] d\mathbf{z},\ \textit{for almost all}\ \ \mathbf{x},\mathbf{y}\in\mathbb{R}^n, \quad (6.12)$$

and the nonlocal divergence operator $\mathcal{D}$ in terms of $\boldsymbol{\nu}(\mathbf{x},\mathbf{y})$ as [102]

$$\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) = \frac{1}{2}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \left[\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y}) - \boldsymbol{\kappa}(\mathbf{y},\mathbf{x},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{x})\right] d\mathbf{z}d\mathbf{y},\ \textit{for almost all}\ \ \mathbf{x}\in\mathbb{R}^n. \quad (6.13)$$

Now, the expression in Eq. (6.8) is taken as the general definition of the nonlocal divergence operator in [5]. Depending on the choice of kernel, different definitions nonlocal (divergence) operators can be defined [5]. In [102], the kernel is taken to be $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})$, where $\delta(\cdot)$ denotes the Dirac delta function and $\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \in \left[L^1(\mathbb{R}^n\times\mathbb{R}^n)\right]^k$, so that, from Eq. (6.13) we get

$$\begin{aligned}\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) &= \frac{1}{2}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \left[2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y}) - 2\boldsymbol{\alpha}(\mathbf{y},\mathbf{x})\delta(\mathbf{y}-\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{x})\right] d\mathbf{z}d\mathbf{y} \\ &= \int_{\mathbb{R}^n} \left[\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) - \boldsymbol{\alpha}(\mathbf{y},\mathbf{x})\cdot\boldsymbol{\nu}(\mathbf{y},\mathbf{x})\right] d\mathbf{y}.\end{aligned} \quad (6.14)$$

Now, by restricting the function $\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})$ to be antisymmetric (discussions on why $\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})$ should be antisymmetric are provided in [102][5]), Eq. (6.14) can be recast as

$$\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) = \int_{\mathbb{R}^n} \left[\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) + \boldsymbol{\nu}(\mathbf{y},\mathbf{x})\right]\cdot\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})d\mathbf{y}. \quad (6.15)$$

Remarks:

- If we substitute the kernel $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})$ in Eq. (6.8)

$$
\begin{aligned}
\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) &= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} [2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})]\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n} [2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})]\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y})d\mathbf{y} \\
&= \int_{\mathbb{R}^n} [\boldsymbol{\nu}(\mathbf{x},\mathbf{y})]\cdot\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})d\mathbf{y} + \int_{\mathbb{R}^n} [\boldsymbol{\nu}(\mathbf{x},\mathbf{y})]\cdot\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})d\mathbf{y}.
\end{aligned}
\tag{6.16}
$$

We see from Eq. (6.10) and Eq. (6.4) that

$$
\begin{aligned}
\psi(\mathbf{x},\mathbf{y}) &= \int_{\mathbb{R}^n} 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z} \\
&= 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) \\
&= -\psi(\mathbf{y},\mathbf{x}) = -2\boldsymbol{\alpha}(\mathbf{y},\mathbf{x})\cdot\boldsymbol{\nu}(\mathbf{y},\mathbf{x}),
\end{aligned}
\tag{6.17}
$$

which implies

$$
2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) = -2\boldsymbol{\alpha}(\mathbf{y},\mathbf{x})\cdot\boldsymbol{\nu}(\mathbf{y},\mathbf{x}), \tag{6.18}
$$

$$
2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) - 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{y},\mathbf{x}), \tag{6.19}
$$

$$
2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot[\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) - \boldsymbol{\nu}(\mathbf{y},\mathbf{x})] = 0, \tag{6.20}
$$

$$
\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) = \boldsymbol{\nu}(\mathbf{y},\mathbf{x}). \tag{6.21}
$$

Substituting Eq. (6.21) in Eq. (6.16) leads to Eq. (6.15) and

$$
\psi(\mathbf{x},\mathbf{y}) = 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) = [\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) + \boldsymbol{\nu}(\mathbf{y},\mathbf{x})]\cdot\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}), \tag{6.22}
$$

which is consistent with Eqs. (6.15), (6.8) and (6.10).

- In [5], Eq. (6.8) is taken as the basis for the definition of the nonlocal divergence operator. By choosing $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})$, where $\delta(\cdot)$ denotes the Dirac delta function and $\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \in \left[L^1(\mathbb{R}^n \times \mathbb{R}^n)\right]^k$ is a vector-valued function, the nonlocal vector calculus presented in [102][72] is recovered. In fact, from Eq. (6.8), we get

$$\begin{aligned}
\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) &= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} [\delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})] \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} d\mathbf{y} + \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} d\mathbf{y} \\
&= \int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{x},\mathbf{y}) d\mathbf{y} + \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{y} d\mathbf{z} \\
&= \int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{x},\mathbf{y}) d\mathbf{y} + \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{x}) d\mathbf{z} \\
&= \int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{x},\mathbf{y}) d\mathbf{y} + \int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{y},\mathbf{x}) d\mathbf{y} \\
&= \int_{\mathbb{R}^n} \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot [\boldsymbol{\nu}(\mathbf{x},\mathbf{y}) + \boldsymbol{\nu}(\mathbf{y},\mathbf{x})]\, d\mathbf{y},
\end{aligned} \tag{6.23}$$

where in the second term of the third step the order of double integration has been reversed and where the integration variable has been changed from $\mathbf{z}$ to $\mathbf{y}$ in the sixth step. It can be observed that Eq. (6.23) is the same as Eq. (6.15), and it is consistent with Eqs. (6.8) and (6.10) and the flux obtained in Eq. (6.22). However, substituting $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})$ in the definition of flux in Eq. (6.10) leads to

$$\begin{aligned}
\psi(\mathbf{x},\mathbf{y}) &= \int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} \\
&= \int_{\mathbb{R}^n} [\delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})] \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z} \\
&= \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \cdot \boldsymbol{\nu}(\mathbf{x},\mathbf{y}) + \int_{\mathbb{R}^n} [\delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})] \cdot \boldsymbol{\nu}(\mathbf{z},\mathbf{y}) d\mathbf{z}.
\end{aligned} \tag{6.24}$$

As aforementioned, in fact, the process in Eq. (6.23) relies on switching the double integrals and a change of variables, and this is not possible from Eq. (6.10). It can be observed, though, that the kernel employed in [5] (i.e., $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) + \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})$) recovers the one used in [102] (i.e., $\boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z}) = 2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})$), if $\mathbf{y}$ and $\mathbf{z}$ are swapped in the second term. From Eq. (6.23) we get

$$
\begin{aligned}
\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) &= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \boldsymbol{\kappa}(\mathbf{x},\mathbf{y},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} [\delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})+\delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})]\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}+\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}+\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{y})\boldsymbol{\alpha}(\mathbf{x},\mathbf{z})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{y}d\mathbf{z} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}+\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{y},\mathbf{z})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y}+\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} \delta(\mathbf{x}-\mathbf{z})\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y} \\
&= \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} [2\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})\delta(\mathbf{x}-\mathbf{z})]\cdot\boldsymbol{\nu}(\mathbf{z},\mathbf{y})d\mathbf{z}d\mathbf{y},
\end{aligned}
\tag{6.25}
$$

where Eq. (6.21) has been employed.

6.1.1 Nonlocal divergence and gradient

Based on Section 6.1, here we provide the definitions for nonlocal divergence and gradient from [102]. The interested reader is advised to pay careful attention while reading the literature, as different different authors sometimes use different nomenclature, definitions, and symbols for these operators (see for example [72][69]). Given a vector two-point function (i.e., a function defined for pairs of points) $\boldsymbol{\nu}:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^k$ and the antisymmetric vector two-point function $\boldsymbol{\alpha}:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^k$, the nonlocal divergence operator $\mathcal{D}$ on $\boldsymbol{\nu}$, so that $\mathcal{D}\boldsymbol{\nu}:\mathbb{R}^n\to\mathbb{R}$, is defined as

$$
\mathcal{D}\boldsymbol{\nu}(\mathbf{x}) := \int_{\mathbb{R}^n}(\boldsymbol{\nu}+\boldsymbol{\nu}')\cdot\boldsymbol{\alpha}d\mathbf{y},\ \ \mathit{for}\ \ \mathbf{x}\in\mathbb{R}^n, \tag{6.26}
$$

where $\boldsymbol{\nu}=\boldsymbol{\nu}(\mathbf{x},\mathbf{y})$, $\boldsymbol{\nu}'=\boldsymbol{\nu}(\mathbf{y},\mathbf{x})$ and $\boldsymbol{\alpha}=\boldsymbol{\alpha}(\mathbf{x},\mathbf{y})$.

The nonlocal gradient operator $\mathcal{G}$ acting on a scalar two-point function $\eta:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$

is defined, given an antisymmetric vector two-point function $\boldsymbol{\beta} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^k$ as

$$\mathcal{G}\eta(\mathbf{x}) := \int_{\mathbb{R}^n} (\eta + \eta') \cdot \boldsymbol{\beta} d\mathbf{y}, \;\; \textit{for} \;\; \mathbf{x} \in \mathbb{R}^n, \tag{6.27}$$

where $\mathcal{G}\eta : \mathbb{R}^n \to \mathbb{R}^k$. The adjoint operators of the aforementioned nonlocal operators are the two-point operators defined as follows [102]. Given a point function (i.e., a function defined at points) $u : \mathbb{R}^n \to \mathbb{R}$ the adjoint of $\mathcal{D}$ is the two-point operator given by

$$\mathcal{D}^* u(\mathbf{x}, \mathbf{y}) := -(u' - u)\boldsymbol{\alpha}, \;\; \textit{for} \;\; \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{6.28}$$

where $u' = u(\mathbf{y})$ and $u = u(\mathbf{x})$ and $\mathcal{D}^* u : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^k$. Given the vector point function $(v\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$, the adjoint of $\mathcal{G}$ is the two-point operator that acts on $\mathbf{v}$ as:

$$\mathcal{G}^* \mathbf{v}(\mathbf{x}, \mathbf{y}) := -(\mathbf{v}' - \mathbf{v}) \cdot \boldsymbol{\beta}, \;\; \textit{for} \;\; \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{6.29}$$

Now, we can also define the nonlocal divergence operator acting on tensors by applying the definion in Eq. (6.26) to each row of the tensor [102]. Given the tensor, two-point function $\boldsymbol{\Psi} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m \times k}$ and the antisymmetric vector two-point function $\boldsymbol{\alpha} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^k$, the nonlocal point divergence operator $\mathcal{D}_t$ for tensors is defined as

$$\mathcal{D}_t \boldsymbol{\Psi}(\mathbf{x}) := \int_{\mathbb{R}^n} (\boldsymbol{\Psi} + \boldsymbol{\Psi}') \cdot \boldsymbol{\alpha} d\mathbf{y}, \;\; \textit{for} \;\; \mathbf{x} \in \mathbb{R}^n, \tag{6.30}$$

where $\mathcal{D}_t \boldsymbol{\Psi} : \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{\Psi} = \boldsymbol{\Psi}(\mathbf{x}, \mathbf{y})$, and $\boldsymbol{\Psi}' = \boldsymbol{\Psi}(\mathbf{y}, \mathbf{x})$. Now, given the vector point function $\mathbf{v}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$, the adjoint of $\mathcal{D}_t$ is given by [102]

$$(\mathcal{D}_t^* \mathbf{v})(\mathbf{x}, \mathbf{y}) = -(\mathbf{v}' - \mathbf{v}) \otimes \boldsymbol{\alpha}, \;\; \textit{for} \;\; \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{6.31}$$

where $\mathcal{D}_t^* \mathbf{v} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m \times k}$, $\mathbf{v} = \mathbf{v}(\mathbf{x})$, and $\mathbf{v}' = \mathbf{v}(\mathbf{y})$. Also, the definition of the nonlocal gradient operator of a two-point scalar function given in Eq. (6.27) can be extended to define the nonlocal gradient operator $\mathcal{G}_t$ acting on a two-point vector function $\boldsymbol{\psi}(\mathbf{x},\mathbf{y})$ as the point tensor function:

$$\mathcal{G}_t\boldsymbol{\psi}(\mathbf{x}) := \int_{\mathbb{R}^n} (\boldsymbol{\psi}' + \boldsymbol{\psi}) \otimes \boldsymbol{\beta} d\mathbf{y}. \tag{6.32}$$

The corresponding adjoint operator $\mathcal{G}_t^*$ acting on a point tensor $\mathbf{U}(\mathbf{x})$ is given by [102]:

$$(\mathcal{G}_t^*\mathbf{U})(\mathbf{x},\mathbf{y}) = -(\mathbf{U}' - \mathbf{U}) \cdot \boldsymbol{\beta}, \tag{6.33}$$

where $\mathbf{U}' = \mathbf{U}(\mathbf{y})$ and $\mathbf{U} = \mathbf{U}(\mathbf{x})$.

6.1.2 Nonlocal interaction operators

Let us start by considering an open subset $\Omega \subset \mathbb{R}^n$; we define the interaction domain corresponding to Ω as [102]:

$$\mathcal{B}\Omega := \{\mathbf{y} \in \mathbb{R}^n \setminus \Omega \; such \; that \; \boldsymbol{\alpha}(\mathbf{x},\mathbf{y}) \neq \mathbf{0} \; for \; some \; \mathbf{x} \in \Omega\}, \tag{6.34}$$

so that $\mathcal{B}\Omega$ consists of those points outside of Ω that interact with points in Ω. At this point, we can define operators that describe the interaction between points in $\mathcal{B}\Omega$ and those in $\overline{\overline{\Omega}} = \Omega \cup \mathcal{B}\Omega$. Following [102] we can define the point interaction operator $\mathcal{N}\boldsymbol{\nu}(\mathbf{x}) := \mathcal{B}\Omega \to \mathbb{R}$, $\forall \mathbf{x} \in \mathcal{B}\Omega$,

$$\mathcal{N}\boldsymbol{\nu}(\mathbf{x}) := -\int_{\Omega \cup \mathcal{B}\Omega} (\boldsymbol{\nu} + \boldsymbol{\nu}') \cdot \boldsymbol{\alpha} d\mathbf{y}, \tag{6.35}$$

which corresponds to the nonlocal divergence operator defined in Eq. (6.26). Corresponding to the point gradient operator in Eq. (6.27) we can also define the point interaction operator $\mathcal{S}\eta : \mathcal{B}\Omega \to \mathbb{R}^k$, $\forall \mathbf{x} \in \mathcal{B}\Omega$, as

$$\mathcal{S}\eta(\mathbf{x}) := -\int_{\Omega \cup \mathcal{B}\Omega} (\eta + \eta')\boldsymbol{\beta} d\mathbf{y}. \tag{6.36}$$

6.1.3 Nonlocal integral theorems

Based on the notations and definitions presented in Sections 6.1.1 and 6.1.2 the following theorems can be proved [102][72]

$$\int_{\Omega} \mathcal{D}\boldsymbol{\nu} d\mathbf{x} = \int_{\mathcal{B}\Omega} \mathcal{N}\boldsymbol{\nu} d\mathbf{x}, \tag{6.37}$$

$$\int_{\Omega} \mathcal{G}\eta d\mathbf{x} = \int_{\mathcal{B}\Omega} \mathcal{S}\eta d\mathbf{x}. \tag{6.38}$$

Eq. 6.37 is referred to as the nonlocal Gauss theorem [72]. Also, nonlocal integration by parts formulas can be defined [102], which are analogue to the local ones. Given the point functions $u(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{v}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^k$, we have [102]

$$\int_{\Omega} u\mathcal{D}\boldsymbol{\nu} d\mathbf{x} - \int_{\Omega\cup\mathcal{B}\Omega}\int_{\Omega\cup\mathcal{B}\Omega} \mathcal{D}^* u \cdot \boldsymbol{\nu} d\mathbf{y} d\mathbf{x} = \int_{\mathcal{B}\Omega} u\mathcal{N}\boldsymbol{\nu} d\mathbf{x}, \tag{6.39}$$

$$\int_{\Omega} \mathbf{v} \cdot \mathcal{G}\eta d\mathbf{x} - \int_{\Omega\cup\mathcal{B}\Omega}\int_{\Omega\cup\mathcal{B}\Omega} \mathcal{G}^* \mathbf{v}\eta d\mathbf{y} d\mathbf{x} = \int_{\mathcal{B}\Omega} \mathbf{v} \cdot \mathcal{S}\eta d\mathbf{x}. \tag{6.40}$$

6.2 Variationally consistent integration for high order exactness in weak form peridynamics

6.2.1 Model problem

The concept of variationally consistent integration for high order exactness was introduced in [61] for boundary value problems of the following form:

$$\begin{aligned} -Lu &= b, \quad \text{in } \Omega \\ u &= g, \quad \text{on } \partial\Omega_g \\ Bu &= h, \quad \text{on } \partial\Omega_h \end{aligned} \quad , \tag{6.41}$$

where L is a generic differential operator acting in $\Omega \subset \mathbb{R}^d$, b is a source term, g is the prescribed values of u on the essential zero-volume boundary $\partial\Omega_g$., B is a boundary operator acting on the Neumann boundary $\partial\Omega_h$, and the domain boundary $\partial\Omega$ is such that $\partial\Omega = \partial\Omega_g \cup \partial\Omega_h$ and $\partial\Omega_g \cap \partial\Omega_h = \emptyset$. Here, we extend it to boundary value problems involving integral operators, such as the peridynamic equation of motion. We therefore consider the following nonlocal constrained value problem:

$$\begin{aligned} -\overline{\Lambda}\mathbf{u} &= \mathbf{b}, \quad \text{in } \Omega \\ \mathbf{u} &= \mathbf{g}, \quad \text{on } \mathcal{B}\Omega_g \ , \\ \overline{\mathcal{N}}\mathbf{u} &= \mathbf{h}, \quad \text{on } \mathcal{B}\Omega_h \end{aligned} \tag{6.42}$$

where $\overline{\mathcal{N}}$ is a linear operator of volume constraints on the non-zero volume $\mathcal{B}\Omega_h$ and $\overline{\Lambda}$ is a nonlocal integral operator. If $\overline{\Lambda}\mathbf{u} = \mathcal{D}_t(\beta\mathcal{D}_t^*(\mathbf{u})^T)$ with $\mathbf{u} \in \mathbb{R}^d$ and β being a chosen kernel, for example, corresponds to the linear operator of bond based peridynamics [69][102]. When $d = 1$ the nonlocal Laplacian operator $\mathcal{L}u = \mathcal{D}(\beta\mathcal{D}^*(u)^T)$ is recovered. For simplicity, in the rest of the discussion we consider

$$\begin{aligned} -\mathcal{L}u &= b, \quad \text{in } \Omega \\ u &= g, \quad \text{on } \mathcal{B}\Omega_g \ \cdot \\ \mathcal{N}(\beta\mathcal{D}^*u) &= h, \quad \text{on } \mathcal{B}\Omega_h \end{aligned} \tag{6.43}$$

6.2.2 Integration constraints

Now, let us consider the nonlocal constrained value problem in Eq.6.43 where the solution is complete polynomials with degree n:

$$u = \sum_{|\alpha| \le n} \mathbf{c}\mathbf{x}^{\alpha} = u^n, \tag{6.44}$$

where the multi-index notation introduced in Chapter 3 has been used. The corresponding problem with consistent force term and boundary conditions is

$$
\begin{aligned}
-\mathcal{L}u^n &= b, \quad \text{in } \Omega \\
u^n &= g, \quad \text{on } \mathcal{B}\Omega_g \ , \\
\mathcal{N}(\beta \mathcal{D}^* u^n) &= h, \quad \text{on } \mathcal{B}\Omega_h
\end{aligned}
\tag{6.45}
$$

that is, when the source terms and boundary conditions in Eq. (6.45) are specified, the solution is Eq. (6.44). Similarly to what can be done for the classical theory, by multiplying the first equation in the strong formulation in Eq. (6.45) by a test function $v \in \mathbb{R}$ (with $v = 0 \in \mathcal{B}\Omega$) and by applying the nonlocal integration by parts in Eq. (6.39), a variational (weak) formulation for the problem in Eq. (6.43) can be obtained:

$$a(v,u)_\Omega = (v,b)_\Omega + (v,h)_{\mathcal{B}\Omega_h}, \tag{6.46}$$

where

$$a(v,u)_\Omega = \int_{\Omega \cup \mathcal{B}\Omega} \int_{\Omega \cup \mathcal{B}\Omega} \mathcal{D}^* v \beta \mathcal{D}^* u dy dx \tag{6.47}$$

$$(v,b)_\Omega = \int_\Omega v b dx, \tag{6.48}$$

$$(v,h)_{\mathcal{B}\Omega_h} = \int_{\mathcal{B}\Omega_h} v h dx, \tag{6.49}$$

which for Eq. (6.45) becomes:

$$a(v,u^n)_\Omega = (v,b)_\Omega + (v,h)_{\mathcal{B}\Omega_h}, \tag{6.50}$$

Now, let us consider a Galerkin approximation where we approximate the trial and test funcions u and v with u^h and v^h, respectively, making Eq. 6.51

$$a(v^h,u^h)_\Omega = (v^h,b)_\Omega + (v^h,h)_{\mathcal{B}\Omega_h}. \tag{6.51}$$

Now, let

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{J=1}^{NP} \Psi_J(\mathbf{x}) d_J, \tag{6.52}$$

and

$$v(\mathbf{x}) \approx v^h(\mathbf{x}) = \sum_{I=1}^{NP} \overline{\Psi}(\mathbf{x})_I v_I, \tag{6.53}$$

where

$$\overline{\Psi}_I(\mathbf{x}) = \begin{cases} \Psi_I(\mathbf{x}), & \text{for } \mathbf{x} \in \overline{\overline{\Omega}} \setminus \mathscr{B}\Omega_g \\ 0, & \text{for } \mathbf{x} \in \mathscr{B}\Omega_g \end{cases}. \tag{6.54}$$

Now, let us consider the $n-th$ order completeness of u^h [61]:

$$u^n = \sum_{J=1}^{NP} \Psi_J d_J^n, \tag{6.55}$$

where u^n is the complete $n-th$ order polynomial defined in Eq. (6.44), and

$$d_J^n = \sum_{|\alpha| \leq n} \mathbf{c}_\alpha \mathbf{x}_J^\alpha, \; \forall \mathbf{x} \in \Omega. \tag{6.56}$$

In a meshfree setting, with NP discretization nodes, the $n-th$ order completeness is easily satisfied by using the reproducing kernel particle method and associated shape functions. By taking $d_J = d_J^n$ in Eq. (6.52) and by substituting Eqs. (6.52) and (6.53) into Eq. (6.51), we get:

$$\sum_{J=1}^{NP} a(\overline{\Psi}_I, \Psi_J d_J^n)_\Omega = (\overline{\Psi}_I, b)_\Omega + (\overline{\Psi}_I, h)_{\mathcal{B}\Omega_h}, \; \forall I. \tag{6.57}$$

Now considering all orders of completeness from 1 to n and employing numerical integration in Eq. (6.57) we get

$$a\langle \overline{\Psi}_I, \mathbf{x}^\alpha \rangle_\Omega = -\langle \overline{\Psi}_I, \mathcal{L}\mathbf{x}^\alpha \rangle_\Omega + \langle \overline{\Psi}_I, \mathcal{N}(\beta \mathcal{D}^* \mathbf{x}^\alpha) \rangle_{\mathcal{B}\Omega_h}, \; \forall I, \; |\alpha| = 0, 1 ..., n, \tag{6.58}$$

where $a\langle \cdot, \cdot \rangle_\Omega$ is the numerical quadrature version of $a(\cdot, \cdot)_\Omega$, and $\langle \cdot, \cdot \rangle_\Omega$ and $\langle \cdot, \cdot \rangle_{\mathcal{B}\Omega_h}$ are the numerical quadrature versions of $(\cdot,\cdot)_{\Omega}$ and $(\cdot,\cdot)_{\mathcal{B}\Omega_h}$, respectively. The equations in Eq. (6.58) are called the $n-th$ order integration constraints and when satisfied yield to $n-th$ order Galerkin exactness.

6.2.3 Integration constraints for the nonlocal linear bond-based steady-state peridynamic problem

For illustration, let us consider the one-dimensional linear bond-based peridynamic problem described by

$$\begin{cases} -\int_{x-\delta}^{x+\delta} c\omega(|\xi|)(u(x')-u(x))dx' = b(x), & \text{for } x \in \Omega \\ u(x) = g(x), & \text{for } x \in \mathscr{B}\Omega_g \\ -\int_{x-\delta}^{x+\delta} c\omega(|\xi|)(u(x')-u(x))dx' = h(x), & \text{for } x \in \mathscr{B}\Omega_h \end{cases}, \tag{6.59}$$

where Ω is the interior of the domain, and $x \in \mathscr{B}\Omega_g$ and $x \in \mathscr{B}\Omega_h$ are the Dirichlet and Neumann boundaries, respectively. By multiplying by $v(x) \in V$ (where $v(x) = 0$, $\forall x \in \mathscr{B}\Omega_g$) and by employing the nonlocal version of integration by parts, the weak form of Eq. (6.59) can be obtained:

$$\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[v(x')-v(x)]c\omega(|\xi|)[u(x')-u(x)]dx'dx = \int_{\Omega} v(x)b(x)dx + \int_{\mathscr{B}\Omega_h} v(x)h(x)dx. \tag{6.60}$$

Now, we introduce the approximations in Eq. (6.52) and Eq. (6.53) for the trial and test functions, respectively, and obtain

$$\begin{aligned} &\sum_{J=1}^{NP}\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[\overline{\Psi}_I(x')-\overline{\Psi}_I(x)]c\omega(|\xi|)[\Psi_J(x')-\Psi_J(x)]dx'dxd_J \\ &= \int_{\Omega}\overline{\Psi}_I(x)b(x)dx + \int_{\mathscr{B}\Omega_h}\overline{\Psi}_I(x)h(x)dx,\ \forall I. \end{aligned} \tag{6.61}$$

Let us insert $d_J = d_J^n$, the associated body force b^n and Neumann boundary conditions h^n, and use the completeness condition (see Eq. (6.55)):

$$\begin{aligned}&\sum_{J=1}^{NP}\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[\overline{\Psi}_I(x')-\overline{\Psi}_I(x)]c\omega(|\xi|)[\Psi_J(x')-\Psi_J(x)]dx'dxd_J^n\\&=\int_{\Omega}\overline{\Psi}_I(x)b^n(x)dx+\int_{\mathscr{B}\Omega_h}\overline{\Psi}_I(x)h^n(x)dx,\ \forall I,\end{aligned}\tag{6.62}$$

$$\begin{aligned}&\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[\overline{\Psi}_I(x')-\overline{\Psi}_I(x)]cw(|\xi|)[u^n(x')-u^n(x)]dx'dx\\&=\int_{\Omega}\overline{\Psi}_I(x)b^n(x)dx+\int_{\mathscr{B}\Omega_h}\overline{\Psi}_I(x)h^n(x)dx,\ \forall I,\end{aligned}\tag{6.63}$$

$$\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[\overline{\Psi}_I(x')-\overline{\Psi}_I(x)]c\omega(|\xi|)[u^n(x')-u^n(x)]dx'dx=\int_{\Omega\cup\mathscr{B}\Omega_h}\overline{\Psi}_I(x)\tilde{b}^n(x)dx,\ \forall I,\tag{6.64}$$

where

$$\tilde{b}^n(x)=\begin{cases}b^n(x), & \text{for } x\in\Omega\\ h^n(x), & \text{for } x\in\mathscr{B}\Omega_h\end{cases},\tag{6.65}$$

where we recall that $b^n(x)$ and $h^n(x)$ are the body force and Neumann boundary conditions associated with the $n-th$ order polynomial solution $u^n(x)$, respectively. Considering all orders of completeness from 1 to n and employing numerical integration in Eq. (6.64), we obtain the following integration constraints:

$$\frac{1}{2}\hat{\int}_{\overline{\overline{\Omega}}}\hat{\int}_{\overline{\overline{\Omega}}}[\overline{\Psi}_I(x')-\overline{\Psi}_I(x)]c\omega(|\xi|)[x'^s-x^s]dx'dx=\hat{\int}_{\Omega\cup\mathscr{B}\Omega_h}\overline{\Psi}_I(x)\tilde{b}^s(x)dx,\ \forall I,\ s=1,2,...,n,\tag{6.66}$$

where, from Eq. (6.59),

$$\tilde{b}^s(x)=-\int_{x-\delta}^{x+\delta}c\omega(|\xi|)(x'^s-x^s)dx',\ \forall x\in\Omega\cup\mathscr{B}\Omega_h.\tag{6.67}$$

6.3 Node-based variationally consistent integration method: symmetrical weights

In this section, we propose a node-based variationally consistent integration method for the weak form of peridynamics. We illustrate it on the problem presented in Section 6.2.3. Node-based approaches are simple and computationally efficient compared to other methods, and therefore well-suited for engineering applications, but are limited in integration accuracy. We consider a discretization of NP nodes. The peridynamic weak form in Eq. (6.60), nodally integrated becomes

$$\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[v(x_P)-v(x_L)]c\omega(x_L,x_P)[u(x_P)-u(x_L)]V_PV_L=\sum_{Q=1}^{\overline{NP}}v(x_Q)\tilde{b}(x_Q)V_Q, \tag{6.68}$$

where $\omega(x_L,x_P)=\omega(|\xi|)=\omega(|x_P-x_L|)$, $\tilde{b}$ is defined as in Eq. (6.65), and NP and $\overline{NP}$ are the nodes in $\overline{\overline{\Omega}}$ and $\Omega\cup\mathscr{B}\Omega_h$, respectively. It can be noted that the contribution of P and L to the left-hand side of Eq. (6.69) is symmetrical, i.e.,

$$[v(x_P)-v(x_L)]c\omega(x_L,x_P)[u(x_P)-u(x_L)]V_PV_L=[v(x_L)-v(x_P)]c\omega(x_P,x_L)[u(x_L)-u(x_P)]V_LV_P. \tag{6.69}$$

Now, by introducing the approximations in Eq. (6.52) and Eq. (6.53) for the trial and test functions in Eq. (6.69), we get

$$\sum_{J=1}^{NP}\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}_I(x_P)-\overline{\Psi}_I(x_L)]c\omega(x_L,x_P)[\Psi_J(x_P)-\Psi_J(x_L)]V_PV_Ld_J=\sum_{Q=1}^{\overline{NP}}\overline{\Psi}_I(x_Q)\tilde{b}(x_Q)V_Q,\ \forall I, \tag{6.70}$$

which, by taking $d = d^n$ and using the n-th order completeness of the shape functions (see Eq. (6.55)), becomes

$$\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}_I(x_P)-\overline{\Psi}_I(x_L)]c\omega(x_L,x_P)[u^n(x_P)-u^n(x_L)]V_PV_L=\sum_{Q=1}^{\overline{NP}}\overline{\Psi}_I(x_Q)\tilde{b}^n(x_Q)V_Q,\ \forall I. \tag{6.71}$$

Satisfaction of the integration constraints in Eq. (6.66) for the node-based integration approach requires:

$$\begin{aligned}&\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}_I(x_P)-\overline{\Psi}_I(x_L)]c\omega(x_L,x_P)[(x_P^s-x_L^s)]V_PV_L\\&=\sum_{Q=1}^{\overline{NP}}\overline{\Psi}_I(x_Q)\tilde{b}^s(x_Q)V_Q,\ \forall I,\ s=1,2,...,n,\end{aligned} \tag{6.72}$$

$$\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}_I(x_P)-\overline{\Psi}_I(x_L)]c\omega(x_L,x_P)[x_P^s-x_L^s]V_{PL}=\sum_{Q=1}^{\overline{NP}}\overline{\Psi}_I(x_Q)\tilde{b}^s(x_Q)V_Q,\ \forall I,\ s=1,2,...,n, \tag{6.73}$$

where $V_{PL}=V_PV_L=V_{LP}$. One way of satisfying Eq. (6.72) is to replace the integration weights V_{PL} with modified ones:

$$V_{PL}^g=\sum_{r=1}^{n}\alpha_r(|x_P-x_L|)^rV_PV_L. \tag{6.74}$$

It is clear from the definition in Eq. (6.67) that $V_{PL}^g=V_{LP}^g$, i.e. the symmetry of interaction between P and L is retained. This leads, $\forall I$, to the following system of equations where $s=1,...,n$:

$$\sum_{r=1}^{n}\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}_I(x_P)-\overline{\Psi}_I(x_L)]c\omega(x_L,x_P)(x_P^s-x_L^s)\alpha_r(|x_P-x_L|)^rV_PV_L=\sum_{Q=1}^{\overline{NP}}\overline{\Psi}_I(x_Q)\tilde{b}^s(x_Q)V_Q, \tag{6.75}$$

which can be rewritten as:

$$\mathbf{A}^{\alpha}\boldsymbol{\alpha}=\mathbf{b}, \tag{6.76}$$

where

$$A_{sr}^{\alpha}=\frac{1}{2}\sum_{L=1}^{NP}\sum_{P=1}^{NP}[\overline{\Psi}(x_P)-\overline{\Psi}(x_L)]c\omega(x_L,x_P)(x_P^s-x_L^s)(|x_P-x_L|)^rV_PV_L, \tag{6.77}$$

$$\boldsymbol{\alpha}^T = [\alpha_1, ..., \alpha_n] \tag{6.78}$$

and

$$\mathbf{b}^T = [\sum_{Q=1}^{\overline{NP}} \overline{\Psi}_I(x_Q)\tilde{b}^1(x_Q)V_Q, ..., \sum_{Q=1}^{\overline{NP}} \overline{\Psi}_I(x_Q)\tilde{b}^n(x_Q)V_Q]. \tag{6.79}$$

As long as $\mathbf{A}^\alpha$ is invertible, $\boldsymbol{\alpha}$ can be found and the modifications to the weights V_{PL} that guarantee the n-th order integration constraint to be satisfied can be found. Lastly, it can be argued that, by modifying the integration weights for the given node-based integration scheme, we are indirectly scaling and modifying the test functions in the weak form, similarly to the spirit of the variationally consistent integration methods proposed in [61] for PDEs.

6.4 Numerical example: one-dimensional static peridynamic problem

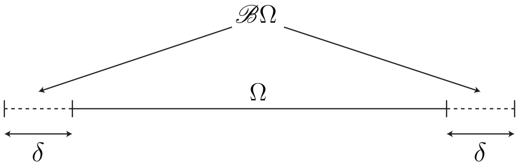


Figure 6.1: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega_g$

Consider a unit length one-dimensional domain (see Figure 7.8) $\overline{\overline{\Omega}} = [0,1]$, with inner domain $\Omega = (\delta, 1-\delta)$ and boundary layer $\mathscr{B}\Omega_g = [0,\delta] \cup [1-\delta, 1]$, and a one-dimensional force state.

$$\underline{\mathrm{T}}[x,t]\langle\xi\rangle = \frac{1}{2}cw(|\xi|)(u(x') - u(x)), \tag{6.80}$$

where $\xi = x' - x$. The associated static linear bond-based peridynamic problem is

$$\begin{cases} -\int_{x-\delta}^{x+\delta} cw(|\xi|)(u(x') - u(x))dx' = b(x), & \text{for } x \in \Omega \\ u(x) = g(x), & \text{for } x \in \mathscr{B}\Omega_g \end{cases} \tag{6.81}$$

where $w(|\xi|)$ is the influence function, $g(x)$ is a prescribed Dirichlet boundary condition function, and c is a constitutive constant. In this example, the influence function is taken to be Boolean:

$$w(|\xi|) = \begin{cases} 1, & \text{for } |\xi| \le \delta \\ 0, & \text{for } |\xi| > \delta \end{cases} \tag{6.82}$$

and the constitutive constant c is taken as

$$c = \frac{2K}{m}, \tag{6.83}$$

where K is a constant and m is the weighted volume [18][24][95]:

$$m = \int_{-\delta}^{\delta} w(|\xi|)\xi^2 d\xi \tag{6.84}$$

Manufactured solutions of polynomial form up to third-order are considered, i.e. $u(x) = k_3x^3 + k_2x^2 + k_1x + k_0$ with k_3, k_2, k_1, and k_0 being constant coefficients. The body force density for Eq. (6.81) is then

$$\begin{aligned} b(x) &= -\int_{x-\delta}^{x+\delta} cw(|\xi|)[k_3(x'^3 - x^3) + k_2(x'^2 - x^2) + k_1(x' - x)]dx' \\ &= -\int_{-\delta}^{\delta} cw(|\xi|)[k_3((x+\xi)^3 - x^3) + k_2((x+\xi)^2 - x^2) + k_1\xi]d\xi \\ &= -2K(3k_3x + k_2), \end{aligned} \tag{6.85}$$

The weak form associated with Eq. (6.81):

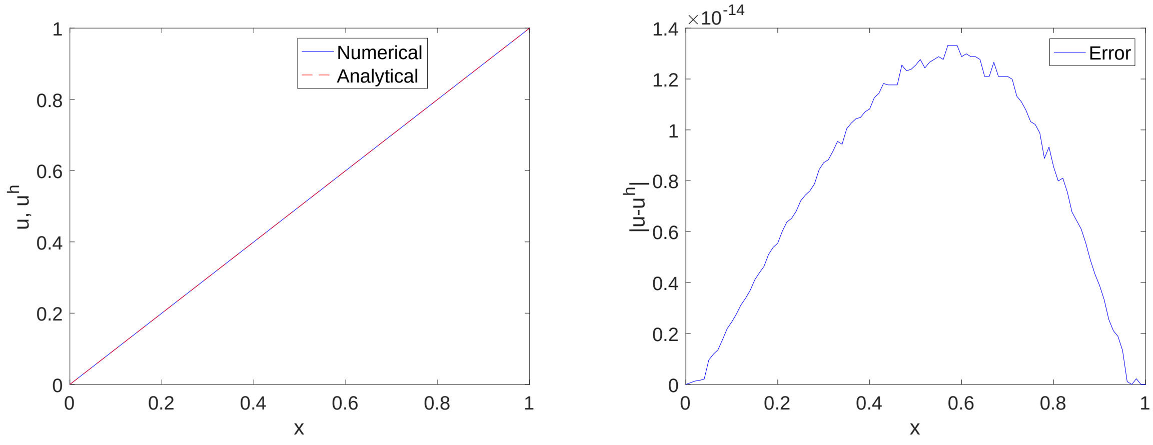
$$\frac{1}{2}\int_{\overline{\overline{\Omega}}}\int_{\overline{\overline{\Omega}}}[v(x')-v(x)]c\omega(x,x')[u(x')-u(x)]dx'dx = \int_{\Omega} v(x)b(x)dx, \tag{6.86}$$

where $v(x) \in V$, $v = 0 \in \mathscr{B}\Omega_g$.

6.4.1 Uniform discretization

First, we solve the static linear problem with manufactured polynomial solution (see Eq. (6.81)) by considering a uniform discretization and an RKPM approximation. The discretization contains $N = 100$ equally spaced nodes with spacing h in the problem domain interval $[0,1]$. We consider three problems: one with manufactured linear solution, one with a quadratic solution and one with a cubic solution. First-order, second-order, and third-order bases for the RK approximation are used for each problem, respectively. A cubic B-spline kernel was selected in all cases, with support sizes $a = 1.5h$, $2.5h$, and $3.5h$, respectively. We select an horizon $\delta = 0.04$.

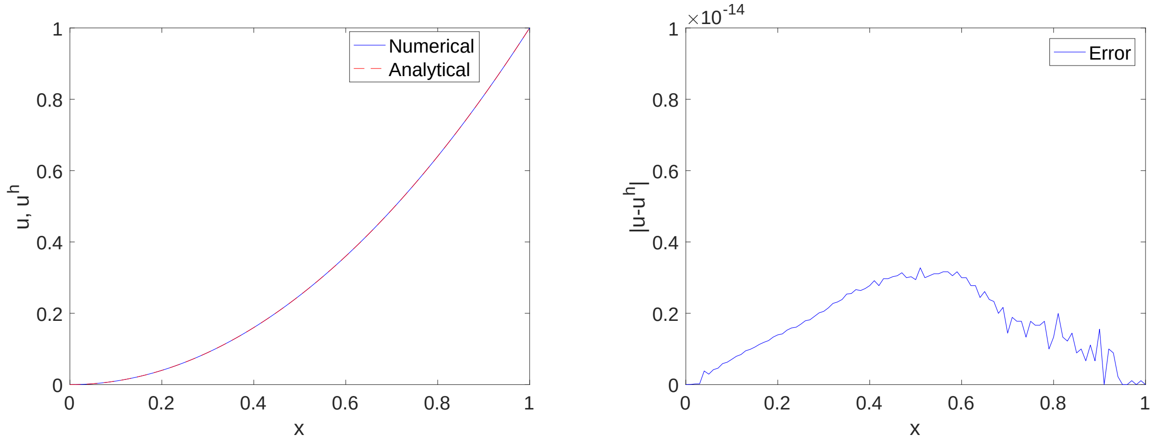
It has to be noted that for a uniform discretization, the linear integration constraint is automatically satisfied for symmetrical weights. Furthermore, it was noticed that, though scaled, the equation imposing third-order variational consistency was the same equation (i.e, linearly dependent) as the one imposing the second-order integration constraint, making $\mathbf{A}^{\alpha}$ in Eq. (6.76) singular. For this reason, for the uniform discretization we reduce the system in Eq. (6.75) by only considering $r = s = 2$. As shown in Figures 6.2, 6.3, and 6.4, these integration weights satisfy the first, second and third-order integration constraints, respectively.



(a) First-order consistency

(b) Absolute error

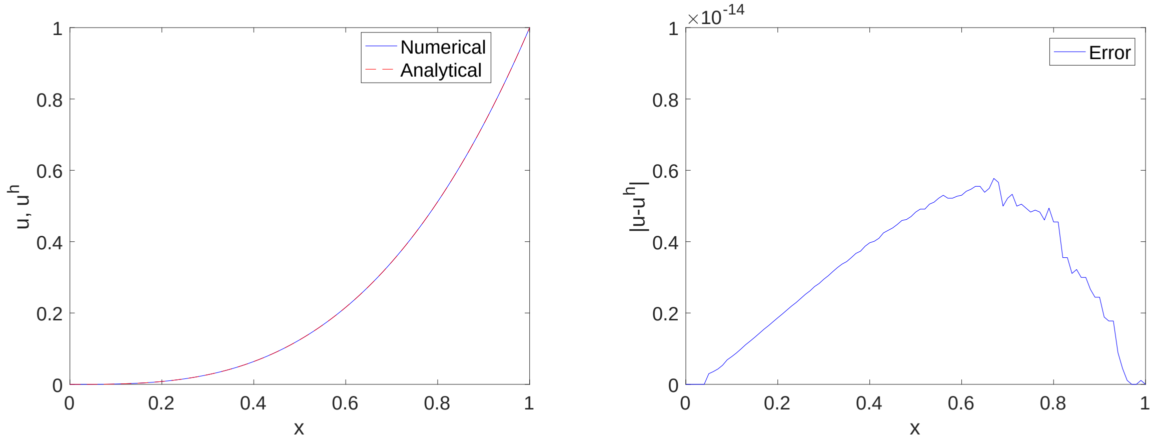
Figure 6.2: First-order variational consistency condition obtained with imposition of second-order integration constraint, linear RK shape functions and a uniform point distribution



(a) Second-order consistency

(b) Absolute error

Figure 6.3: Second-order variational consistency condition obtained with imposition of second-order integration constraint, quadratic RK shape functions and a uniform point distribution



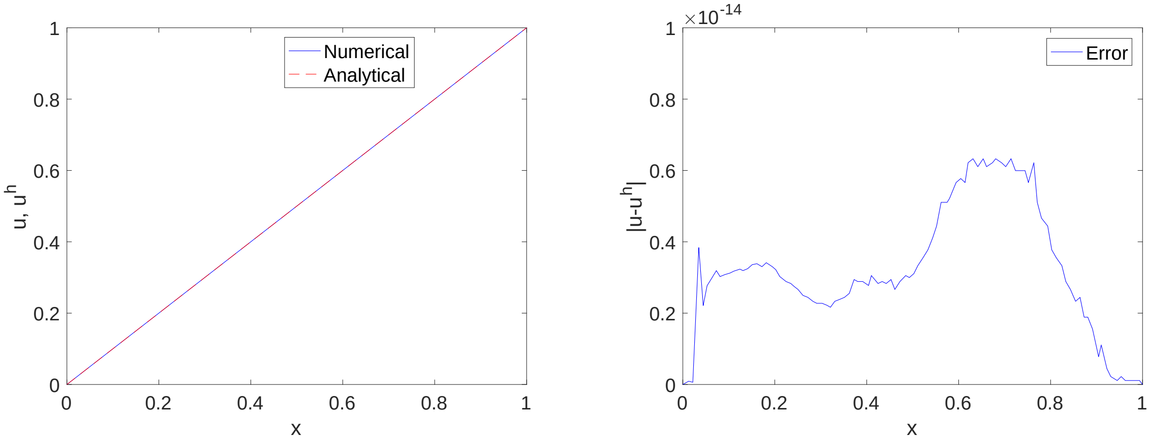
(a) Third-order consistency

(b) Absolute error

Figure 6.4: Third-order variational consistency condition obtained with imposition of second-order integration constraint, cubic RK shape functions and a uniform point distribution

6.4.2 Non-uniform discretization

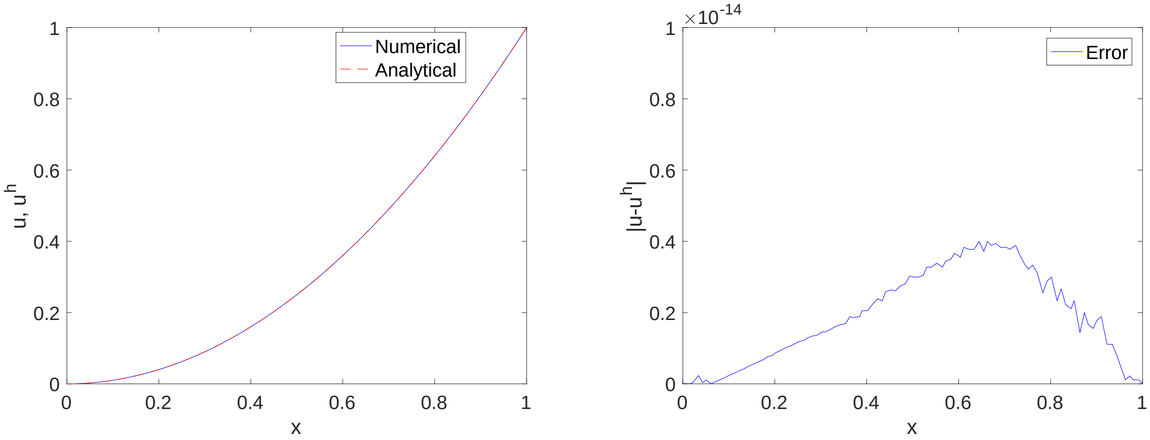
Here, the behaviour of the proposed node-based variationally consistent integration approach is investigated for a non-uniform discretization. We solve again the static linear problem with manufactured polynomial solution (see Eq. (6.81)). The non-uniform discretization is obtained by perturbing a uniform discretization with $N = 100$ nodes (and corresponding spacing of size h) in the problem domain interval $[0, 1]$. This was achieved by moving each node in $(0, 1)$ from their original position x_I in the uniform grid to a new randomly selected position $x_I^{nu} \in [x_I - \varepsilon h, x_I + \varepsilon h]$, where ε is a perturbation factor, selected randomly in $[0, 0.5]$. As for the uniform case, we consider problems with manufactured linear, quadratic and cubic solution, and used first-order, second-order, and third-order basis for the RK approximation, respectively. A cubic B-spline kernel was selected in all cases, with support sizes $a = 1.5h$, $2.5h$, and $3.5h$, respectively. We select an horizon $\delta = 0.04$. For the non-uniform case, the first-order, second-order and third-order integration variational consistency conditions are linearly independent and can therefore be imposed simultaneously. Integration constraints up to first, second, and third-order are imposed for the problems with manufactured linear, quadratic, and cubic solutions, respectively. Figures 6.5, 6.6, and 6.7, show that the node-based integration, modified to satisfy the chosen integration constraints, leads to recovering the exact solution for the considered problems.



(a) First-order consistency

(b) Absolute error

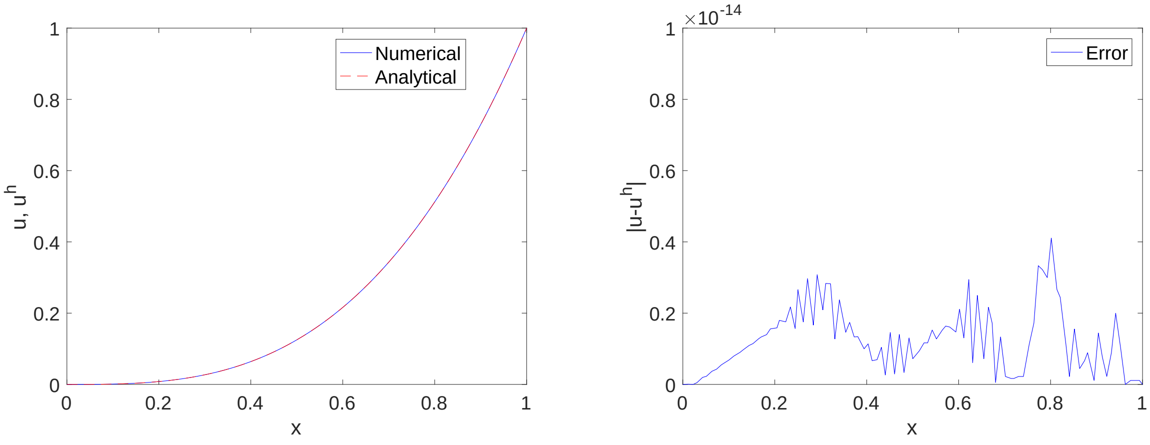
Figure 6.5: First-order variational consistency condition obtained with imposition of first-order integration constraint, linear RK shape functions and a non-uniform point distribution.



(a) Second-order consistency

(b) Absolute error

Figure 6.6: Second-order variational consistency condition obtained with imposition of second-order integration constraint, quadratic RK shape functions and a non-uniform point distribution.



(a) Third-order consistency

(b) Absolute error

Figure 6.7: Third-order variational consistency condition obtained with imposition of third-order integration constraint, cubic RK shape functions and a non-uniform point distribution.

Acknowledgments

This chapter is currently being prepared for submission for publication of the material. M. Pasetto, J.S. Chen, “Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics”. The dissertation author was the primary investigator of this material.

Chapter 7

A Waveform Relaxation Newmark Method for Transient RKPM Modelling of Dynamic Problems

7.1 The Waveform Relaxation scheme

The Waveform Relaxation scheme is an iterative method used to solve systems of time dependent differential equations. To briefly review the key ideas of the WR method we start by considering a dynamical system that can be described as a set of p second-order ordinary differential equations, with associated initial conditions, of the form:

$$\begin{aligned} \ddot{\mathbf{d}} &= \mathbf{f}(t,\mathbf{d}) \quad t \in [0,T], \\ \mathbf{d}(0) &= \mathbf{d}_0, \\ \dot{\mathbf{d}}(0) &= \mathbf{v}_0, \end{aligned} \tag{7.1}$$

where $T > 0$ is the final simulation time, $\mathbf{f} : [0,T] \times \mathbb{R}^p \to \mathbb{R}^p$ is a generalized load vector, $\mathbf{d}(t) = [d_1(t)\, d_2(t) \ldots d_p(t)]^t \in \mathbb{R}^p$ is the displacement vector, and $\mathbb{R}^p$ denotes a real valued vector space of dimension p. $\mathbf{d}_0 = [d_{1,0}\, d_{2,0} \ldots d_{p,0}]^t \in \mathbb{R}^p$ and $\mathbf{v}_0 = [v_{1,0}\, v_{2,0} \ldots v_{p,0}]^t \in \mathbb{R}^p$ are vectors which contain the initial displacements $d_{i,0}$ and initial velocity values $v_{i,0}$, respectively.

The key idea of the WR algorithm is to (i) uncouple the set of equations, (ii) integrate efficiently the uncoupled set of equations in time, (iii) repeat the process with the additional information generated from the previous integration step until convergence is reached. Equation decoupling in step (i) is achieved through an assignment-partitioning process, while steps (ii)-(iii) consist of an iterative relaxation procedure. In the assignment-partitioning process, each unknown variable is assigned to one equation of the original system (7.1), which is then partitioned into disjoint subsystems consisting of one (point-wise decomposition) or more (block-wise decomposition) equations [68]. Therefore, within each subsystem there are the assigned internal variables to be solved for and external variables, which are internal variables of the other subsystems. Each subsystem is then solved iteratively over the time domain during the relaxation process. In order to accelerate convergence and reduce storage, the time domain can be divided into intervals called windows. The solution of each subsystem can be performed independently, even on different processors, and the information is exchanged once the solution over the considered time window has taken place [41].

Figure 7.1 illustrates the underlying WR concept for a 1D space-time domain. The space domain is first discretized into independent function variables (indicated by the dots) and assigned an arbitrary value over the time window at iteration $\nu = 0$; each subsystem is then solved iteratively by using the information coming from the other subsystems (external variables) until convergence (dashed line) for its assigned internal variable is achieved. The values of the external variables get updated using the information from the current or previous iterations, depending on the chosen decoupling WR scheme [68]. As shown in Figure 7.1, once convergence is reached in one time window the obtained result is then used as a starting point for the successive window. In Figure 7.1, the continuous lines represent the solutions obtained for each iteration ν while the dashed lines are the converged solutions; the dot points represent the values at the end of each time window, taken as initial conditions for the subsequent one.

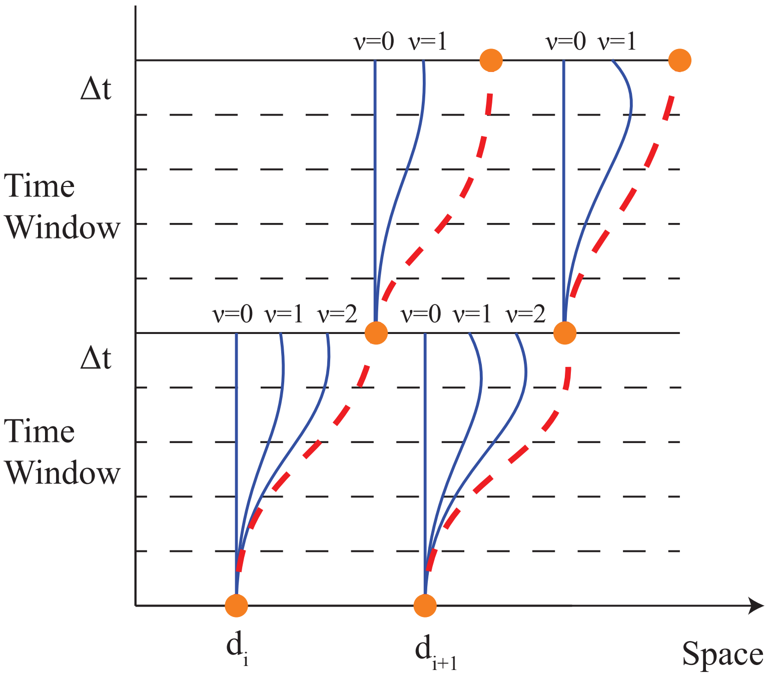


Figure 7.1: Space decomposition and iterations over the time windows.

In this work we will limit ourselves to the two most commonly used types of relaxation schemes: the Gauss-Seidel (GS) WR and the Jacobi WR. When the Gauss-Seidel relaxation is used, the uncoupled equations are solved sequentially and the waveform solution of one decomposed subsystem is used immediately as input to update the approximate waveforms of the other subsystems. The point-wise GS iteration scheme can therefore be expressed as

$$\begin{cases} \ddot{d}_i^{(\nu+1)} = f_i(t, d_1^{(\nu+1)}, \ldots, d_{i-1}^{(\nu+1)}, d_i^{(\nu+1)}, d_{i+1}^{(\nu)}, \ldots, d_p^{(\nu)}) \\ d_i^{(\nu+1)}(0) = d_{0_i} \\ \dot{d}_i^{(\nu+1)}(0) = v_{0_i} \end{cases}, \tag{7.2}$$

where the superscripts ν and $\nu+1$ denote the iteration count and p as before is the number of unknown variables. Figure 7.2 shows the flow of information if a Gauss-Seidel WR scheme is used.

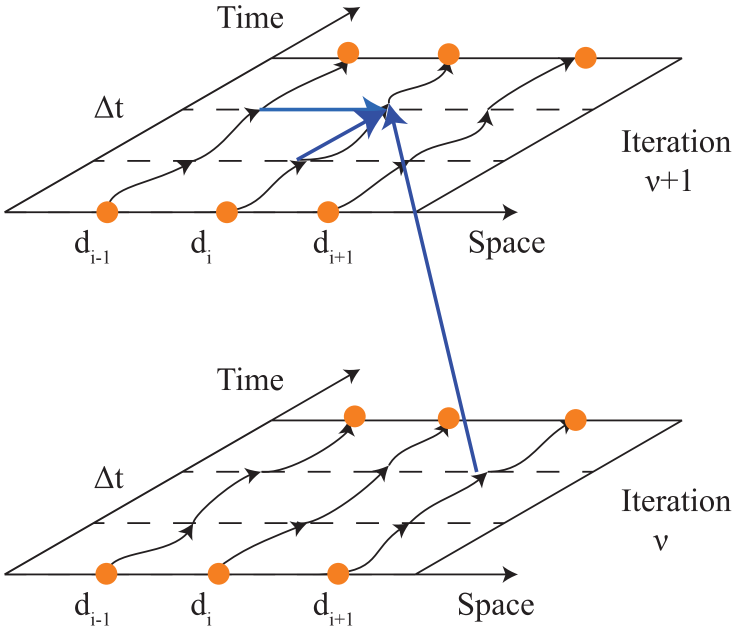


Figure 7.2: Information flow for the Gauss-Seidel WR scheme

Instead, if the Jacobi WR is used, all the uncoupled equations could be solved simultaneously using the information on the external variables available from the previous iteration ν, updated at the beginning of each iteration [68]. The point-wise Jacobi WR scheme can be mathematically expressed as

$$\begin{cases} \ddot{d}_i^{(\nu+1)} = f_i(t, d_1^{(\nu)}, \ldots, d_{i-1}^{(\nu)}, d_i^{(\nu+1)}, d_{i+1}^{(\nu)}, \ldots, d_p^{(\nu)}) \\ d_i^{(\nu+1)}(0) = d_{0_i} \\ \dot{d}_i^{(\nu+1)}(0) = v_{0_i} \end{cases} . \tag{7.3}$$

The flow of information for the Jacobi WR is illustrated in Figure 7.3.

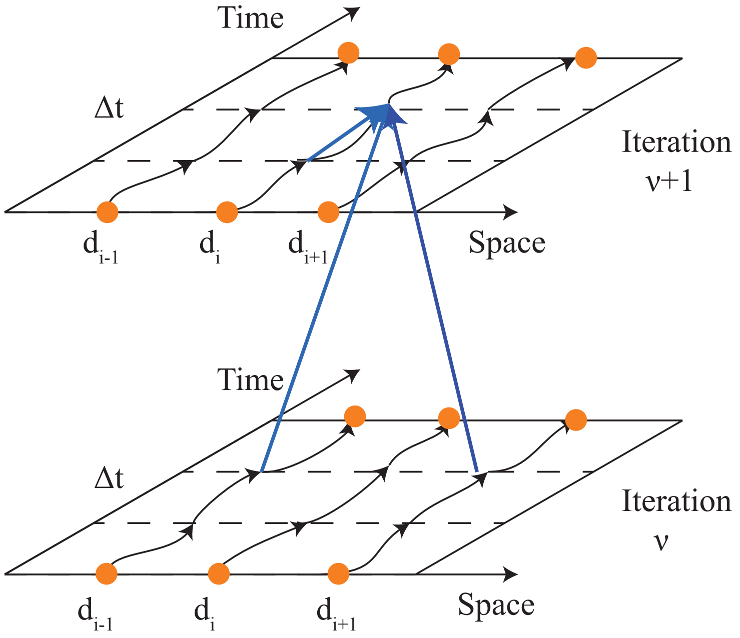


Figure 7.3: Information flow for the Jacobi WR scheme

In addition, a block-wise schemes can also be defined in a similar way by considering multiple equations for each subsystem. For example, a two block Jacobi scheme can be expressed as [131]

$$\begin{cases} \ddot{d}_{2i-1}^{(\nu+1)} = f_i(t, d_1^{(\nu)}, \ldots, d_{2i-2}^{(\nu)}, d_{2i-1}^{(\nu+1)}, d_{2i}^{(\nu+1)}, \ldots, d_p^{(\nu)}) \\ \ddot{d}_{2i}^{(\nu+1)} = f_i(t, d_1^{(\nu)}, \ldots, d_{2i-2}^{(\nu)}, d_{2i-1}^{(\nu+1)}, d_{2i}^{(\nu+1)}, \ldots, d_p^{(\nu)}) \\ d_{2i-1}^{(\nu+1)}(0) = d_{0_{2i-1}} \\ \dot{d}_{2i-1}^{(\nu+1)}(0) = v_{0_{2i-1}} \\ d_{2i}^{(\nu+1)}(0) = d_{0_{2i}} \\ \dot{d}_{2i}^{(\nu+1)}(0) = v_{0_{2i}} \end{cases} . \tag{7.4}$$

Both Jacobi and Gauss-Seidel relaxation schemes are carried out until satisfactory convergence is achieved and are started with an initial approximation $d^{(0)}(t)$ defined over the considered time interval. For known initial conditions, this initial approximation is usually taken to be constant over time and equal to the values specified by the initial conditions [131] so that

$$d_i^{(0)}(t) = d_{0_i}, \qquad \forall t \in [0,T], \qquad i = 1,...,p. \tag{7.5}$$

The Jacobi type approach allows more parallelization at the expense of slower convergence rates with respect to the Gauss-Seidel WR scheme [41].

As mentioned in Chapter 2, the WR method was first developed as a continuous-time scheme. In practical applications though, the time domain is usually discretized and the differential equations composing the WR subsystems are solved by using conventional numerical integration methods [68][131]; in this study the solution is performed by means of a standard Newmark-β integration scheme, resulting in the proposed WR-Newmark method (WRN$_\beta$).

In the system defined by Eq. (7.1), the equation coupling was present through the d_i variables while the $\ddot{d}_i$ terms were already uncoupled. In the remaining of this section we present the approach of the WR method for linear systems describing damped structural dynamic problems. General linear structural dynamic problems are governed by a linear second order hyperbolic system of ODEs, which is written in the well-known standard semi-discrete form as

$$\begin{cases} \mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f} \\ \mathbf{d}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) = \mathbf{v}_0 \end{cases} , \tag{7.6}$$

where $\mathbf{d}$ is the displacement vector, $\mathbf{f}$ is the vector of applied forces, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are respectively the mass, damping and stiffness matrices, and a dot denotes a time-derivative.

The key idea is to split the mass matrix $\mathbf{M}$ into matrices $\mathbf{M}_+$ and $\mathbf{M}_-$, the damping matrix into $\mathbf{C}_+$ and $\mathbf{C}_-$ and the stiffness matrix $\mathbf{K}$ into $\mathbf{K}_+$ and $\mathbf{K}_-$ so that $\mathbf{M} = \mathbf{M}_+ - \mathbf{M}_-$, $\mathbf{C} = \mathbf{C}_+ - \mathbf{C}_-$

and $\mathbf{K} = \mathbf{K}_+ - \mathbf{K}_-$ and to then consider the following iteration form:

$$\begin{cases} \mathbf{M}_+\ddot{\mathbf{d}}^{(\nu+1)} + \mathbf{C}_+\dot{\mathbf{d}}^{(\nu+1)} + \mathbf{K}_+\mathbf{d}^{(\nu+1)} = \mathbf{M}_-\ddot{\mathbf{d}}^{(\nu)} + \mathbf{C}_-\dot{\mathbf{d}}^{(\nu)} + \mathbf{K}_-\mathbf{d}^{(\nu)} + \mathbf{f} \\ \mathbf{d}^{(\nu+1)}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}^{(\nu+1)}(0) = \mathbf{v}_0 \end{cases} . \tag{7.7}$$

The chosen splitting for the matrices influences the convergence and the computational complexity of the solution [131]; based on the splittings proposed in the literature for systems of first-order linear differential equations, we assume $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ to be decomposed as $\mathbf{L}+\mathbf{D}+\mathbf{U}$, where $\mathbf{D}$ is a diagonal matrix (block diagonal if a block-wise partitioning is used) and $\mathbf{L}$ and $\mathbf{U}$ are respectively strictly lower and strictly upper triangular matrices [131]. The splitting corresponding to the WR Jacobi and WR Gauss-Seidel schemes are as follows:

- Jacobi:	$\mathbf{M}_+ = \mathbf{D}_\mathbf{M}$	$\mathbf{M}_- = -(\mathbf{L}_\mathbf{M} + \mathbf{U}_\mathbf{M})$
	$\mathbf{C}_+ = \mathbf{D}_\mathbf{C}$	$\mathbf{C}_- = -(\mathbf{L}_\mathbf{C} + \mathbf{U}_\mathbf{C})$
	$\mathbf{K}_+ = \mathbf{D}_\mathbf{K}$	$\mathbf{K}_- = -(\mathbf{L}_\mathbf{K} + \mathbf{U}_\mathbf{K})$
- Gauss-Seidel:	$\mathbf{M}_+ = \mathbf{L}_\mathbf{M} + \mathbf{D}_\mathbf{M}$	$\mathbf{M}_- = -\mathbf{U}_\mathbf{M}$
	$\mathbf{C}_+ = \mathbf{L}_\mathbf{C} + \mathbf{D}_\mathbf{C}$	$\mathbf{C}_- = -\mathbf{U}_\mathbf{C}$
	$\mathbf{K}_+ = \mathbf{L}_\mathbf{K} + \mathbf{D}_\mathbf{K}$	$\mathbf{K}_- = -\mathbf{U}_\mathbf{K}$

It has to be noted that if lumped mass matrices are considered no splitting of $\mathbf{M}$ is required. Once the desired splitting has been performed, the system in Eq. (7.7) is integrated over the chosen time interval $t \in [t_0, t_n]$ using the Newmark algorithm. The WR iteration is then terminated when the error between two successive iterations is smaller than a specified tolerance

$$\max\{\left\|\mathbf{d}^{\nu+1}(t) - \mathbf{d}^{\nu}(t)\right\|\} \leq \varepsilon, \tag{7.8}$$

for some small positive positive constant ε. Convergence of the WR method to the free-vibration displacement of a two mass system is shown in Figure 7.4, for illustration purposes.

The procedure of the WR-Newmark method is summarized in tables corresponding to algorithms 1 and 2. As for the rest of this paper, bold capital letters indicate matrices while bold lower case letters indicate vectors. Bold capital letters followed by (t) (as in $\mathbf{X}_d(t)$) indicate a column in a matrix corresponding to discrete time t. Waveforms are stored in $\mathbf{X}_d$, $\mathbf{X}_v$ and $\mathbf{X}_a$, which denote respectively the space-time displacement, velocity and acceleration matrices. Δt is the time integration time step while β and γ are the parameters of the Newmark method. **d**, **v** and **a** indicate the displacement, velocity and acceleration vectors, respectively. The $Splitting$ function corresponds to matrix splitting as described earlier in this section. It can be noted that, if the Jacobi WR scheme is used, the solution step in line 10 of Algorithm 2 does not require any matrix solve and is thus performed at the cost of an explicit scheme even when the overall scheme is unconditionally stable (as traditional Newmark implicit schemes).

Lastly, Algorithm 1 and 2 describe the procedure for a chosen time step Δt. However, in the WRN$_\beta$ method each subsystem or groups of subsystems in the WRN$_\beta$ method can be solved using a different time step (hence Algorithm 2 would be called separately for each subsystem). As mentioned in the introduction and in Section **??**, though, this requires the introduction of an interpolatory scheme.

Remark: Note that the WR-Newmark method is well suited for parallel implementation. The key issues with parallelization correspond to two main aspects: (i) frequency of processors communication and (ii) solver parallelization algorithms. Below we compare these two aspects for standard Newmark and WR-Newmark algorithms.

- A standard Newmark method is said to be an inherently sequential algorithm, since communication between processors must be done after every time step and the algorithm cannot proceed in time without exchange of information between processors. On the other hand, a WR-Newmark algorithm is said to be time-parallel, since each subsystem coming from the

WR assignment-partitioning process can be solved independently on different processors as it is decoupled from all the other subsystems. Hence, communication between processors does not need to be performed after each time step. This can speed up the solution and be especially advantageous when different time steps are used for different subsystems. However, the optimal time-window size needs to be determined by considering the cost of each WR iteration versus the cost for processor communication. This corresponds to the time loop, lines 5-16 in Algorithm 2.

- In case of a standard Newmark method a parallel solver is required to solve the set of equations at every time step. While the systems that arise in linear structural dynamics are sparse and symmetric, scalable parallel solvers are needed to solve these systems efficiently. On the other hand, the WR-Newmark method with Jacobi type splitting does not require any solvers and the solution of the system is obtained automatically. This corresponds to the solver phase, line 10 in Algorithm 2.

Algorithm 5 Waveform Relaxation Newmark scheme

```
$[\mathbf{K},\mathbf{C},\mathbf{M},\mathbf{f},\mathbf{d}_0,\mathbf{v}_0,\mathbf{a}_0,t_0,t_n,\Delta t,\beta,\gamma] = setup()$
$\mathbf{FX} = [t_0,t_n,\beta,\gamma]$
$\mathbf{X}_d \leftarrow [\mathbf{d}_0,...,\mathbf{d}_n]$                                   ▷ initialize on space-time
$\mathbf{X}_v \leftarrow [\mathbf{v}_0,...,\mathbf{v}_n]$
$\mathbf{X}_a \leftarrow [\mathbf{a}_0,...,\mathbf{a}_n]$
$[\mathbf{M}_+,\mathbf{M}_-,\mathbf{C}_+,\mathbf{C}_-,\mathbf{K}_+,\mathbf{K}_-] = Splitting(\mathbf{M},\mathbf{C},\mathbf{K})$
$\mathbf{S}_+ = [\mathbf{M}_+,\mathbf{C}_+,\mathbf{K}_+], \mathbf{S}_- = [\mathbf{M}_-,\mathbf{C}_-,\mathbf{K}_-]$
while $NormE \geq \varepsilon$ do
    $\mathbf{X}_d^0 = \mathbf{X}_d$
    $\mathbf{X}_v^0 = \mathbf{X}_v$
    $\mathbf{X}_a^0 = \mathbf{X}_a$
    $[\mathbf{X}_d,\mathbf{X}_v,\mathbf{X}_a] = NK_\beta(\mathbf{S}_+,\mathbf{S}_-,\mathbf{f},\mathbf{X}_d,\mathbf{X}_v,\mathbf{X}_a,\mathbf{FX},\Delta t)$
    $NormE \leftarrow \max_t \{\|\mathbf{X}_d - \mathbf{X}_d^0\|\}$                          ▷ compute residual
end while
```

Algorithm 6 Newmark method (a-form)

```
function $[\mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a] = NK_\beta(\mathbf{S}_+, \mathbf{S}_-, \mathbf{f}, \mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a, \mathbf{FX}, \Delta t)$
$\mathbf{d} \leftarrow \mathbf{d}_0$                                                   ▷ initial conditions
$\mathbf{v} \leftarrow \mathbf{v}_0$
$\mathbf{a} \leftarrow \mathbf{a}_0$
for $t = t_1 : \Delta t : t_n$ do
    $\hat{\mathbf{d}} = \mathbf{d} + \Delta t \mathbf{v} + \frac{\Delta t^2}{2}(1 - 2\beta)\mathbf{a}$                       ▷ predictor phase
    $\hat{\mathbf{v}} = \mathbf{v} + \Delta t(1 - \gamma)\mathbf{a}$
    $\mathbf{f}_{WR} = \mathbf{M}_- \mathbf{X}_a(t) + \mathbf{C}_- \mathbf{X}_v(t) + \mathbf{K}_- \mathbf{X}_d(t)$           ▷ solution phase
    $\bar{\mathbf{f}} = \mathbf{f} + \mathbf{f}_{WR} - \mathbf{K}_+ \hat{\mathbf{d}} - \mathbf{C}_+ \hat{\mathbf{v}}$
    $(\mathbf{M}_+ + \gamma \Delta t \mathbf{C}_+ + \beta \Delta t^2 \mathbf{K}_+)\mathbf{a} = \bar{\mathbf{f}}$                   ▷ solve for $\mathbf{a}$
    $\mathbf{d} = \hat{\mathbf{d}} + \beta \Delta t^2 \mathbf{a}$                                    ▷ corrector phase
    $\mathbf{v} = \hat{\mathbf{v}} + \gamma \Delta t \mathbf{a}$
    $\mathbf{X}_d(t) = \mathbf{d}$                                         ▷ update waveforms
    $\mathbf{X}_v(t) = \mathbf{v}$
    $\mathbf{X}_a(t) = \mathbf{a}$
end for
end function
```

In the following subsection a simple undamped two-mass free-vibration problem is presented along with the results obtained with the time-continuous WR scheme.

Example problem: Free vibration of spring-mass system with two degrees of freedom

To illustrate the key ideas of the Waveform Relaxation method, we begin by considering the following system with two discrete masses and springs. The governing system of equations for this problem is

$$\begin{cases} m_1 \ddot{d_1} + (k_1 + k_2) d_1 - k_2 d_2 = 0 \\ m_2 \ddot{d_2} - k_2 d_1 + k_2 d_2 = 0 \end{cases} , \tag{7.9}$$

with $d_1(0) = d_{1,0}$, $d_2(0) = d_{2,0}$, $\dot{d_1}(0) = v_{1,0}$ and $\dot{d_2}(0) = v_{2,0}$. The exact analytical solution to Eq. (7.9) can be found in Appendix 7-A. Applying the Jacobi and the Gauss-Seidel WR schemes we obtain the following iterative systems:

- Jacobi:

$$\begin{cases} m_1\ddot{d}_1^{(\nu+1)} + (k_1+k_2)d_1^{(\nu+1)} = k_2 d_2^{(\nu)} \\ m_2\ddot{d}_2^{(\nu+1)} + k_2 d_2^{(\nu+1)} = k_2 d_1^{(\nu)} \end{cases} \tag{7.10}$$

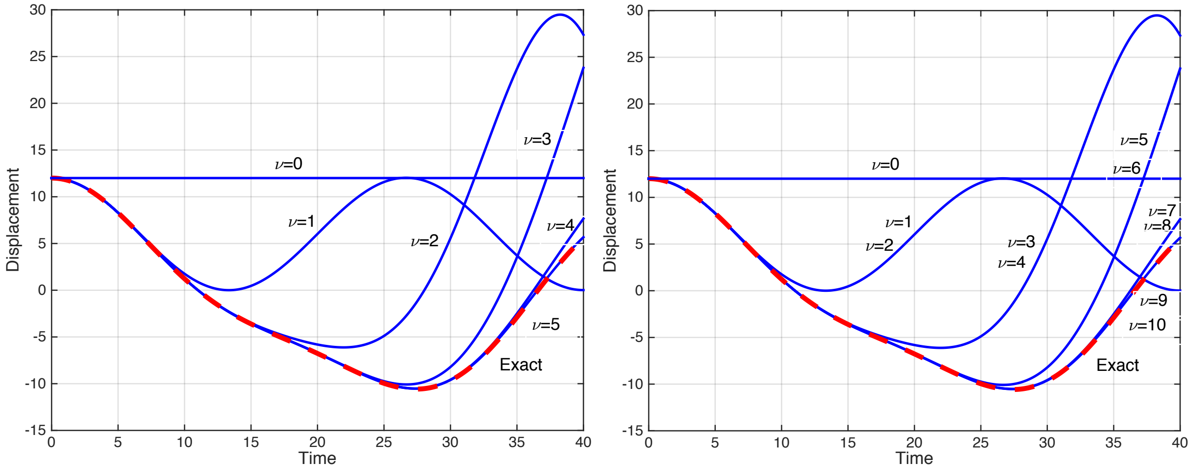
- Gauss-Seidel:

$$\begin{cases} m_1\ddot{d}_1^{(\nu+1)} + (k_1+k_2)d_1^{(\nu+1)} = k_2 d_2^{(\nu)} \\ m_2\ddot{d}_2^{(\nu+1)} + k_2 d_2^{(\nu+1)} = k_2 d_1^{(\nu+1)} \end{cases} \tag{7.11}$$

with the initial conditions

$$\begin{cases} d_i^{(\nu+1)}(0) = d_{0_i} \\ \dot{d}_i^{(\nu+1)}(0) = v_{0_i} \end{cases} .$$

In this example, the initial waveform corresponding to iteration $\nu = 0$ was taken to be constant over time and equal to the values specified by the initial conditions. Note that in this illustrative example the waveform relaxation method is solved analytically with continuous time functions reported in Appendix 7-B. Figure 7.4a and Figure 7.4b show the first five iterations for the continuous Gauss-Seidel WR solution and the first ten iterations for the continuous Jacobi WR solution, respectively, along with the exact analytical solution for the system in Eq. (7.9). It can be observed that convergence is slower for the Jacobi WR scheme with respect to the Gauss-Seidel WR. As noted in [52], for systems with strong coupling each iteration lengthens the time for which the WR solution is close to the exact analytical solution.



(a) Gauss-Seidel WR

(b) Jacobi WR

Figure 7.4: Space-time convergence of Gauss-Seidel and Jacobi waveform relaxation method; the dashed line represents the analytical solutions while the continuous lines represent the WR solution at each iteration ν.

7.2 Stability and convergence analysis

7.2.1 Stability

The stability of the WR iteration is analyzed following the procedure outlined in [45] since each iteration employs a Newmark scheme. Let us start by recalling the split matrix system for the undamped case defined in Eq. (7.7):

$$\mathbf{M}_{+}\ddot{\mathbf{d}}^{(\nu+1)} + \mathbf{K}_{+}\mathbf{d}^{(\nu+1)} = \mathbf{F}, \tag{7.12}$$

where $\mathbf{F} = \mathbf{M}_{-}\ddot{\mathbf{d}}^{(\nu)} + \mathbf{K}_{-}\mathbf{d}^{(\nu)} + \mathbf{f}$.

Next, the system described in Eq. (7.7) is reduced to a single degree of freedom SDOF form through modal decomposition. To this end, we solve the eigenvalue problem associated with Eq. (7.12) for the $\nu+1$ iteration. Since the matrices $\mathbf{M}_{+}$ and $\mathbf{K}_{+}$ do not change for the various iterations, the eigenvalue problem will be the same for every iteration and if one iteration $\nu+1$

is stable, so will the others. Hence, stability of the method is analyzed for a generic iteration $\nu+1$ and the associated superscript will be omitted. The eigenvalue problem to be considered is therefore:

$$(\mathbf{K}_{+} - (\omega_l^{h+})^2 \mathbf{M}_{+})\mathbf{\Psi}_l = 0, \qquad l \in \{1, 2..., n_{eq}\}, \tag{7.13}$$

with

$$0 \leq (\omega_1^{h+})^2 \leq (\omega_2^{h+})^2 \leq ... \leq (\omega_{n_{eq}}^{h+})^2, \tag{7.14}$$

where ω_l^{h+} and n_{eq} represent the l-th system natural frequency and the number of system degrees of freedom, respectively. Note that the eigenvalues and eigenvectors in Eq. (7.13) are associated with the split WRN$_\beta$ system and not with the original system, which would be considered if the Newmark's method was directly implemented.

In order to analyze the stability of the proposed WR-Newmark method, we follow the procedure in [45] and rewrite the system as a set of first order ODE system as follows:

$$\mathbf{y}_{n+1} = \mathbf{A}\mathbf{y}_n + \mathbf{L}_n \quad , \quad y = \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix}, \tag{7.15}$$

where $\mathbf{A}$ is the amplification factor, given as

$$\mathbf{A} = \mathbf{A}_1^{-1}\mathbf{A}_2, \tag{7.16}$$

$$\mathbf{L}_n = \mathbf{A}_1^{-1} \begin{Bmatrix} \frac{\Delta t^2}{2}[(1-2\beta)F_n + 2\beta F_{n+1}] \\ \Delta t[(1-\gamma)F_n + \gamma F_{n+1}] \end{Bmatrix}, \tag{7.17}$$

$$\mathbf{A}_1 = \begin{bmatrix} 1+\Delta t^2\beta(\omega^{h+})^2 & 0 \\ (\omega^{h+})^2\gamma\Delta t & 1 \end{bmatrix} \;,\; \mathbf{A}_2 = \begin{bmatrix} 1-\frac{\Delta t^2}{2}(\omega^{h+})^2(1-2\beta) & \Delta t \\ -\Delta t(1-\gamma)(\omega^{h+})^2 & 1 \end{bmatrix}, \tag{7.18}$$

where F_n and F_{n+1} are the forcing terms of the modal equations defined at time n and $n+1$

respectively, Δt is the discretized time step and the parameters β and γ determine accuracy, stability and type of the Newmark algorithm. In order for $\mathbf{A}$ to be spectrally stable the following conditions are required:

a) $\rho(\mathbf{A}) \leq 1$, where $\rho(\mathbf{A})$ is the spectral radius of $\mathbf{A}$, defined as
$\rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|$

b) Eigenvalues of $\mathbf{A}$ of multiplicity greater than one are strictly less than one in modulus

The aforementioned conditions on $\mathbf{A}$ lead to the same stability conditions as for the standard Newmark scheme for β, γ and time step Δt, which are given by [45]:

- Unconditional stability:

  $2\beta \geq \gamma \geq \frac{1}{2}$

- Conditional stability:

  $\gamma \geq \frac{1}{2}, 2\beta < \frac{\gamma}{2} \qquad ; \qquad \omega^{h+}\Delta t \leq \Omega_{crit}$

where the critical sampling frequency Ω_{crit} for an undamped system is $\Omega_{crit} = (\frac{\gamma}{2} - \beta)^{-\frac{1}{2}}$.

Several conclusions can be drawn from this analysis: (i) the stability analysis of WRN_β approach is similar to that of traditional Newmark scheme but with the natural frequencies computed using Eq. (7.13) rather than the original mass and stiffness matrices, (ii) the traditional implicit Newmark method requires the solution of systems of equations while the proposed implicit WRN_β does not, i.e. it is an implicit method at the cost of an explicit scheme with several repeated time sweeps (iii) the critical time step in case of traditional conditionally stable Newmark method is more restrictive than the similar (same β and γ parameters) conditionally stable WRN_β.

In other words, a standard Newmark method will require smaller time steps than WR-Newmark to maintain stability. The proof for this last statement follows.

Let us start by considering the eigenvalue problem associated with the use of the traditional Newmark scheme:

$$(\mathbf{K} - \lambda \mathbf{M})\hat{\mathbf{\Phi}} = \mathbf{0}, \tag{7.19}$$

where λ and $\hat{\mathbf{\Phi}}$ are an eigenvalue and an eigenvector of the system.

Recalling the Rayleigh quotient [45], for any arbitrary vector $\mathbf{\Phi}$ the following inequality holds

$$\frac{\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi}}{\mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi}} \leq \lambda_{max}, \tag{7.20}$$

where λ_{max} is the largest eigenvalue associated with Eq. (7.19). The equality holds true only when $\mathbf{\Phi}$ is the eigenvector of Eq. (7.19) associated with the largest eigenvalue.

By considering the Jacobi splitting ($\mathbf{M}_+ = \mathbf{D_M}$, $\mathbf{K}_+ = \mathbf{D_K}$):

$$\frac{\mathbf{\Phi}^T (\mathbf{K}_+ + \mathbf{K}_-) \mathbf{\Phi}}{\mathbf{\Phi}^T (\mathbf{M}_+ + \mathbf{M}_-) \mathbf{\Phi}} = \frac{\mathbf{\Phi}^T \mathbf{K}_+ \mathbf{\Phi} + \mathbf{\Phi}^T \mathbf{K}_- \mathbf{\Phi}}{\mathbf{\Phi}^T \mathbf{M}_+ \mathbf{\Phi} + \mathbf{\Phi}^T \mathbf{M}_- \mathbf{\Phi}} \leq \lambda_{max}. \tag{7.21}$$

Now, let us assume an eigenvector of the WR-Jacobi scheme given in Eq. (7.13), $\mathbf{\Psi}$ that is associated with the largest eigenvalue of the system such that

$$\frac{\mathbf{\Psi}^T \mathbf{K}_+ \mathbf{\Psi}}{\mathbf{\Psi}^T \mathbf{M}_+ \mathbf{\Psi}} = \lambda_{max}^+, \tag{7.22}$$

where λ_{max}^{+} is the largest eigenvalue. Since $\mathbf{M}_{+}$ and $\mathbf{K}_{+}$ are diagonal matrices, i.e.,

$$\mathbf{M}_{+} = \begin{bmatrix} m_{11} & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & m_{n_{eq}n_{neq}} \end{bmatrix}, \mathbf{K}_{+} = \begin{bmatrix} k_{11} & 0 & 0 & 0 \\ 0 & k_{22} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & k_{n_{eq}n_{neq}} \end{bmatrix}, \tag{7.23}$$

$\mathbf{\Psi}$ is such that it is only non-zero in the i-th position corresponding to $\max_i \left(\frac{k_i}{m_i}\right)$. Specifically, $\mathbf{\Psi}^T = [0...1...0]$, where 1 is in the i-th position, which corresponds to $\max_i \left(\frac{k_i}{m_i}\right) = \lambda_{max}^{+}$. Since both $\mathbf{M}_{-}$ and $\mathbf{K}_{-}$ are matrices with zero diagonals,

$$\mathbf{\Psi}^T \mathbf{K}_{-} \mathbf{\Psi} = 0, \tag{7.24}$$

$$\mathbf{\Psi}^T \mathbf{M}_{-} \mathbf{\Psi} = 0. \tag{7.25}$$

Therefore, by choosing $\mathbf{\Phi} = \mathbf{\Psi}$ in Eq. (7.21) we get

$$\frac{\mathbf{\Psi}^T \mathbf{K}_{+} \mathbf{\Psi} + \mathbf{\Psi}^T \mathbf{K}_{-} \mathbf{\Psi}}{\mathbf{\Psi}^T \mathbf{M}_{+} \mathbf{\Psi} + \mathbf{\Psi}^T \mathbf{M}_{-} \mathbf{\Psi}} = \frac{\mathbf{\Psi}^T \mathbf{K}_{+} \mathbf{\Psi}}{\mathbf{\Psi}^T \mathbf{M}_{+} \mathbf{\Psi}} = \lambda_{max}^{+} \leq \lambda_{max}. \tag{7.26}$$

Furthermore, given that $\hat{\mathbf{\Phi}} \neq \mathbf{\Psi}$, the inequality in Eq. (7.26) becomes strict, meaning that

$$\lambda_{max}^{+} < \lambda_{max}. \tag{7.27}$$

Since the largest eigenvalue associated with the WRN_{β} scheme is smaller than the maximum eigenvalue associated with the traditional Newmark problem, the critical time step for stability of the WRN_{β} method is larger than the one for standard Newmark.

Additionally, if instead of a Jacobi partitioning a Gauss-Seidel one is used, both $\mathbf{M}_{+}$ and $\mathbf{K}_{+}$ are lower triangular matrices. The largest eigenvalue associated with Eq. (7.13) is still the maximum ratio between the diagonal values of $\mathbf{K}_+$ and $\mathbf{M}_+$. The same discussion and conclusions therefore hold. Lastly, it can be noted that the derivation above provides an inequality between the maximum eigenvalue and the largest diagonal element of any matrix that can be expressed as $(\mathbf{M}_+)^{-1}\mathbf{K}$, with $\mathbf{M}_+$ being diagonal with positive diagonal entries.

7.2.2 Convergence

Next, we analyze the convergence of the WR-Newmark method by considering a fixed-point in time with successive iterations ν, assuming a stable method for time integration is chosen, as discussed in the previous section. Consider the semi-discrete dynamic equation at some time n+1, neglecting the damping of the system :

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{K}\mathbf{d}_{n+1} = \mathbf{f}_{n+1}. \tag{7.28}$$

Application of the WR splitting leads to

$$(\mathbf{M}_+ - \mathbf{M}_-)\ddot{\mathbf{d}}_{n+1} + (\mathbf{K}_+ - \mathbf{K}_-)\mathbf{d}_{n+1} = \mathbf{f}_{n+1}, \tag{7.29}$$

and by rewriting as an iterative scheme, one gets

$$\mathbf{M}_+\ddot{\mathbf{d}}_{n+1}^{\nu+1} + \mathbf{K}_+\mathbf{d}_{n+1}^{\nu+1} = \mathbf{f}_{n+1} + \mathbf{M}_-\ddot{\mathbf{d}}_{n+1}^{\nu} + \mathbf{K}_-\mathbf{d}_{n+1}^{\nu}. \tag{7.30}$$

Considering the Newmark's predictor-corrector scheme, the predictor phase is

$$\hat{\mathbf{d}}_{n+1}^{\nu+1} = \mathbf{d}_n^{\nu+1} + \Delta t\dot{\mathbf{d}}_n^{\nu+1} + \frac{\Delta t^2}{2}(1-2\beta)\ddot{\mathbf{d}}_n^{\nu+1}, \tag{7.31}$$

$$\hat{\mathbf{v}}_{n+1}^{\nu+1} = \dot{\mathbf{d}}_n^{\nu+1} + \Delta t(1-\gamma)\ddot{\mathbf{d}}_n^{\nu+1}, \tag{7.32}$$

the solution phase becomes

$$(\mathbf{M}_+ + \beta\Delta t^2\mathbf{K}_+)\ddot{\mathbf{d}}^{\nu+1}_{n+1} = \mathbf{f}_{n+1} + \mathbf{K}_-\mathbf{d}^{\nu}_{n+1} + \mathbf{M}_-\ddot{\mathbf{d}}^{\nu}_{n+1} - \mathbf{K}_+\hat{\mathbf{d}}^{\nu+1}_{n+1}, \tag{7.33}$$

and the corrector phase updated from the previous iteration ν is

$$\mathbf{d}^{\nu}_{n+1} = \hat{\mathbf{d}}^{\nu}_{n+1} + \beta\Delta t^2\ddot{\mathbf{d}}^{\nu}_{n+1}, \tag{7.34}$$

leading to

$$(\mathbf{M}_+ + \beta\Delta t^2\mathbf{K}_+)\ddot{\mathbf{d}}^{\nu+1}_{n+1} = \mathbf{f}_{n+1} + (\mathbf{M}_- + \beta\Delta t^2\mathbf{K}_-)\ddot{\mathbf{d}}^{\nu}_{n+1} + \mathbf{K}_-\hat{\mathbf{d}}^{\nu}_{n+1} - \mathbf{K}_+\hat{\mathbf{d}}^{\nu+1}_{n+1}. \tag{7.35}$$

Since we are considering the convergence of the displacement iteration at time $n+1$ between successive iterations ν and $\nu+1$ we rewrite Eq. (7.33) as

$$\mathbf{P}\mathbf{d}^{\nu+1} = \mathbf{Q}\mathbf{d}^{\nu} + \tilde{\mathbf{f}}, \tag{7.36}$$

where the subscript $(n+1)$ has been dropped and where $\mathbf{P} = \mathbf{M}_+ + \beta\Delta t^2\mathbf{K}_+$, $\mathbf{Q} = \mathbf{M}_- + \beta\Delta t^2\mathbf{K}_-$ and $\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{K}_-\hat{\mathbf{d}}^{\nu} - \mathbf{K}_+\hat{\mathbf{d}}^{\nu+1}$, which leads to

$$\ddot{\mathbf{d}}^{\nu+1} = \mathbf{P}^{-1}\mathbf{Q}\ddot{\mathbf{d}}^{\nu} + \mathbf{P}^{-1}\tilde{\mathbf{f}}. \tag{7.37}$$

Eq. (7.36) can further be rewritten in the following form:

$$\ddot{\mathbf{d}}^{\nu+1} = [\mathbf{I} - \mathbf{P}^{-1}(\mathbf{P} - \mathbf{Q})]\ddot{\mathbf{d}}^{\nu} + \mathbf{P}^{-1}\tilde{\mathbf{f}}, \tag{7.38}$$

that leads to

$$\ddot{\mathbf{d}}^{\nu+1} = \ddot{\mathbf{d}}^{\nu} + \mathbf{P}^{-1}[\tilde{\mathbf{f}} - (\mathbf{P} - \mathbf{Q})\mathbf{d}^{\nu}], \tag{7.39}$$

$$\ddot{\mathbf{d}}^{\nu+1} = \ddot{\mathbf{d}}^{\nu} + \mathbf{P}^{-1}\mathbf{r}^{\nu}, \tag{7.40}$$

where $\mathbf{r}^{\nu} = \tilde{\mathbf{f}} - (\mathbf{P} - \mathbf{Q})\ddot{\mathbf{d}}^{\nu}$ is the residual at ν with respect to the system described by Eq. (7.36) at the considered time $n+1$. By denoting with $\ddot{\mathbf{d}}^{*}$ the exact solution to Eq. (7.33), we define $\mathbf{e}^{\nu+1} = \ddot{\mathbf{d}}^{\nu+1} - \ddot{\mathbf{d}}^{*}$ as the error at $\nu+1$ at time $n+1$; Rewriting Eq. (7.40) in terms of the errors, one gets

$$\mathbf{e}^{\nu+1} = \mathbf{e}^{\nu} + \mathbf{P}^{-1}\mathbf{r}^{\nu}, \tag{7.41}$$

$$\mathbf{e}^{\nu+1} = \mathbf{e}^{\nu} + \mathbf{P}^{-1}[-(\mathbf{P} - \mathbf{Q})\mathbf{e}^{\nu}], \tag{7.42}$$

$$\mathbf{e}^{\nu+1} = [\mathbf{I} - \mathbf{P}^{-1}(\mathbf{P} - \mathbf{Q})]\mathbf{e}^{\nu}, \tag{7.43}$$

$$\mathbf{e}^{\nu+1} = \mathbf{P}^{-1}\mathbf{Q}\mathbf{e}^{\nu} = \mathbf{R}\mathbf{e}^{\nu}, \tag{7.44}$$

where $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$ is the error propagation matrix, which relates the errors in two successive iteration steps and therefore relates the error at every iteration $\nu+1$ to the initial one: $\mathbf{e}^{\nu+1} = \mathbf{R}^{\nu+1}\mathbf{e}^{0}$. Iteration (7.36) converges to the true solution if $\lim_{\nu\to\infty} e^{\nu+1} = 0$. The error converges to zero and therefore (7.36) converges to the true solution if and only if all the eigenvalues of $\mathbf{R}$ have magnitudes less than unity [68]:

$$\rho(\mathbf{R}) < 1, \tag{7.45}$$

$$\rho(\mathbf{P}^{-1}\mathbf{Q}) < 1, \tag{7.46}$$

$$\rho[(\mathbf{M}_{+} + \beta\Delta t^2\mathbf{K}_{+})^{-1}(\mathbf{M}_{-} + \beta\Delta t^2\mathbf{K}_{-})] < 1. \tag{7.47}$$

Furthermore, if matrix $\mathbf{P} - \mathbf{Q}$ is diagonally dominant, then the convergence condition will be satisfied [68]. As presented in Section 7.1, depending on the type of chosen WR iteration scheme, different splittings will be performed, resulting in different error propagation matrices **R**. The above derivation is for the a-form of the WR-Newmark method; a similar derivation for the d-form is presented in Appendix 7-C.

7.3 Numerical results

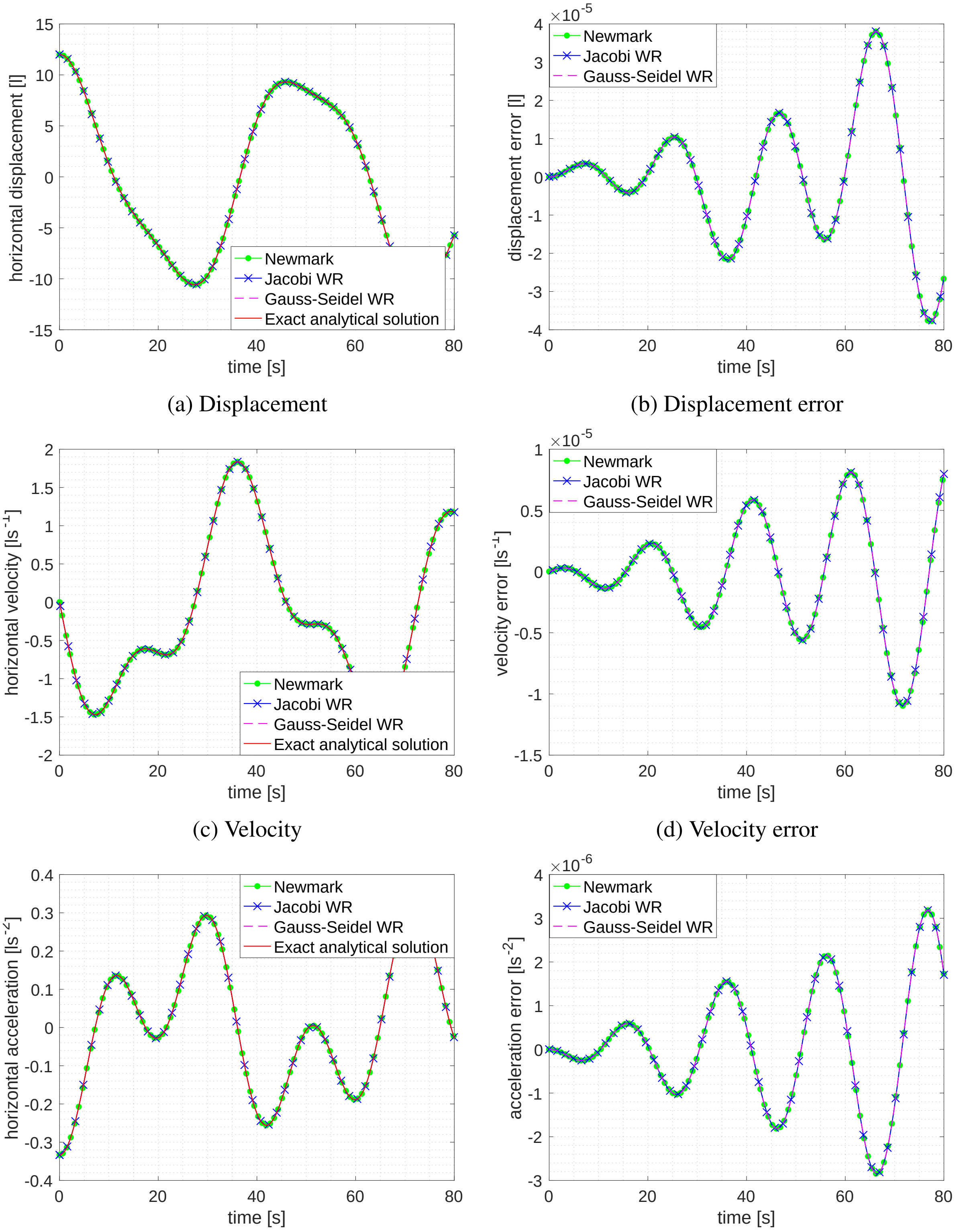
In this section we analyze the performance of the WR-Newmark method on free vibration problems considering the effect of number of degrees of freedom, time integration window and splitting schemes. Four example problems are considered: a two mass-spring system both undamped and damped, a nonlocal one-dimensional wave propagation, and a two dimensional plate subjected to a prescribed initial displacement. In order to show the effectiveness of the proposed algorithm for different structures of mass and stiffness matrices, this last problem is solved not only using an RKPM meshfree method but also a finite element discretization. To assess the accuracy of the algorithm we track the displacements of the nodal points over time and compare them with the analytical solution in the two mass-spring system and with those obtained through the use of the traditional Newmark method for the 2D plate problems. The parameters $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ are used for the time integration and a tolerance $\varepsilon = 10^{-14}$ is used as the iteration convergence criterion.

7.3.1 Free vibration of a two mass-spring system

In this section we consider the free vibration of a two-mass-dashpot-spring system. We begin by the numerical solution of the free vibration of the undamped case that was presented as a model problem in subsection 7.1, for which the analytical solution is known. To assess the accuracy of the WRN_β, Jacobi, and Gauss-Seidel algorithms we track the displacement/velocity/acceleration of mass m_2 over time and compare it with its analytical solution and the traditional Newmark method, as shown in Figure 7.5. The mass matrix **M** for this problem is

diagonal as in the case of lumped mass matrices and $\mathbf{C} = \mathbf{0}$.
An initial displacement $\mathbf{d}_0(t) = (6, 12)^T$ $[m]$ was imposed and the ratios between k_1 and k_2 and m_1 and m_2 were taken respectively as $\frac{k_1}{k_2} = 1$ and $\frac{m_1}{m_2} = 2$ with chosen values: $k_2 = \frac{1}{6}[\frac{N}{m}]$, $m_2 = 3[kg]$. A time step size of $\Delta t = 0.01[s]$ is taken for all integration schemes.

Absolute errors in the displacement, velocity and accelerations solutions of the considered numerical schemes are also reported in Figure 7.5. It can be seen that the error remains quite small on the order of 10^{-5} but increases with the simulation time.



(a) Displacement

(b) Displacement error

(c) Velocity

(d) Velocity error

(e) Acceleration

(f) Acceleration error

Figure 7.5: Undamped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement/velocity/acceleration, $\Delta t = 0.01[s]$; and (right) corresponding absolute error.

Figure 7.6 shows the L_∞ norm of the errors for the different considered schemes for various values of Δt; a convergence rate of 2 with respect to the time step size is observed for all methods.

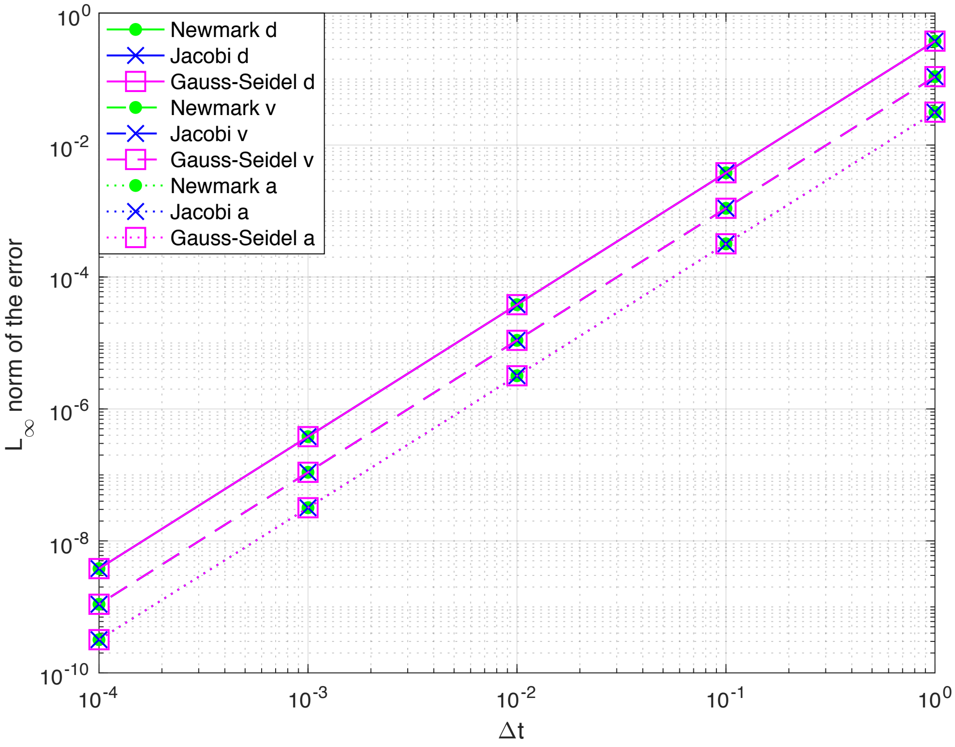


Figure 7.6: L_∞ norm of the error for different values of Δt. The continuous lines represent the error in the displacement solution while the dashed and dotted ones represent the error in the velocity and acceleration, respectively.

Inspection of Figure 7.5 and Figure 7.6 shows that, as expected, the presented WR-Newmark algorithm is comparable to the traditional Newmark method and converges to the exact solution. Note that the rate of convergence for all methods/fields is 2.

Table 7.1 shows the spectral radii of the error propagation matrix **R** for the numerical schemes used. It can be noted that the spectral radius of **R** for the Gauss-Seidel-WRN$_\beta$ is lower with respect to the Jacobi-WRN$_\beta$, indicating a higher convergence rate of the former. This is confirmed by the number of iterations needed for convergence for the two methods for the chosen $\Delta t = 0.01[s]$, reported in the same table. The maximum natural frequency for each of the considered methods is also reported in the Table. It can be seen that the frequency of the standard Newmark method is greater than those of the WR systems.

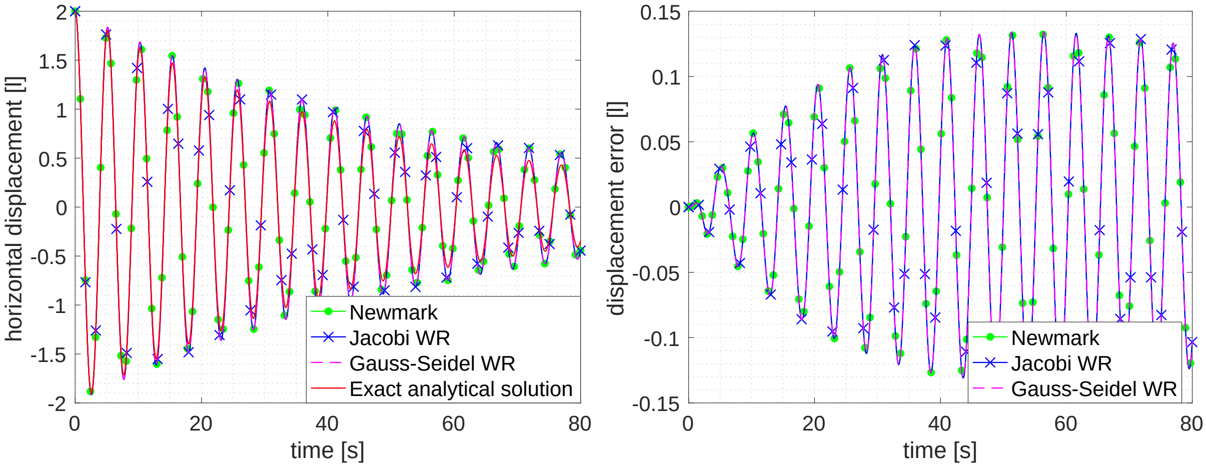
Table 7.1: Spectral radii of the error propagation matrix **R** and number of iterations for for Jacobi WRN$_\beta$ and Gauss-Seidel WRN$_\beta$ and maximum natural frequency for the different methods. $\Delta t = 0.01[s]$.

Method	$\rho(\mathbf{R})$	Number of iterations	$\omega^h_{max}[Hz]$
standard Newmark	-	-	0.31
Jacobi WR	0.9820×10^{-6}	35	0.24
Gauss-Seidel WR	0.9645×10^{-12}	18	0.24

In order to demonstrate the applicability of the proposed method to damped systems, a case with $\mathbf{C} \neq \mathbf{0}$ is also considered. As for the undamped example, the accuracy of the proposed algorithm is assessed through comparison of the numerically obtained displacement of mass m_2 and its analytical solution. An initial displacement $\mathbf{d}_0(t) = (1,2)^T$ $[m]$ was imposed and the ratios between k_1 and k_2 and m_1 and m_2 were taken respectively as $\frac{k_1}{k_2} = 2$ and $\frac{m_1}{m_2} = 2$ with chosen values: $k_2 = 3[\frac{N}{m}]$, $m_2 = 1[kg]$. The damping matrix was taken as

$$\mathbf{C} = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.1 \end{bmatrix} \left[\frac{Ns}{m}\right]$$

The displacement solutions and absolute errors obtained by using the WR-Newmark algorithm with Jacobi and Gauss-Seidel splittings with a time step $\Delta t = 0.01[s]$ are presented in Figure 7.7 together with the traditional Newmark solution and the analytical solution; We note that even for damped systems, the results of the proposed WRN_β method are comparable with those obtained from the traditional Newmark scheme.



(a) Displacement

(b) Displacement error

Figure 7.7: Damped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement; and (right) corresponding absolute error.

Finally, we study the optimal window size in terms of computational cost for all WRN$_\beta$ methods. That is, convergence is achieved in each time window before moving on to the next one. In order to find the optimal time window size, the CPU time necessary to solve the monodimensional two mass problem with the Jacobi and Gauss-Seidel WR-Newmark methods is studied with increasing time window sizes. A time step $\Delta t = 0.01[s]$ and an overall time domain of $120[s]$ are used. The obtained results shown in Table 7.2 suggest that a single time step is the least computationally expensive time-window for WR methods. Hence in subsequent examples we employ a single time step as our choice for time window length.

Table 7.2: CPU time for different time windows dimensions.

Time Window	WR Jacobi CPU $[s]$	WR Gauss-Seidel CPU $[s]$
$1\Delta t$	0.0008	0.0008
$10\Delta t$	0.0012	0.0012
$50\Delta t$	0.021	0.023
$100\Delta t$	0.038	0.04

7.3.2 One-dimensional dynamic linear nonlocal problem

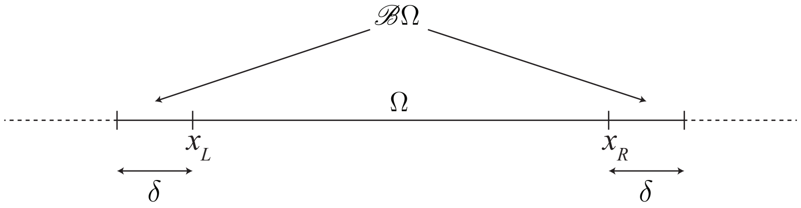


Figure 7.8: 1D domain $\overline{\overline{\Omega}} := \Omega \cup \mathscr{B}\Omega$.

In order to show the applicability of the proposed WRN_β to nonlocal problems, we consider the problem of the relaxation of an initially deformed infinite bar composed of a linear microelastic material presented in [90]. For $x \in \mathbb{R}$, $t \in [0,T]$, the associated initial value problem is

$$\begin{cases} \rho_0 \ddot{u}(x,t) &= \int_{-\infty}^{\infty} C(x'-x)(u(x',t)-u(x,t))dx' + b(x,t) \\ u(x,0) &= u_0(x) \\ \dot{u}(x,0) &= v_0(x) \end{cases}, \tag{7.48}$$

where $b(x,t)$ is the externally applied force density, $C(x'-x)$ is the material micromodulus function, and $u_0(x)$ and $v_0(x)$ are the initial displacement and velocity, respectively. The chosen initial conditions are

$$u_0(x) = Ue^{-\left(\frac{x}{L}\right)^2}, \quad v_0(x) = 0, \tag{7.49}$$

while the micromodulus function $C(x'-x)$ is taken to be the following Gaussian function:

$$C(x'-x) = \frac{4E}{l^3\sqrt{\pi}} e^{-\left(\frac{x'-x}{l}\right)^2}, \tag{7.50}$$

where E is the Young's modulus of the material in classical linear elasticity, l is a material length-scale parameter that captures the effect of long-range forces, L is a length-scale associated with the initial conditions and $\lambda = l/L$.

For $b(x,t)=0$, the exact solution for the problem in Eq. (7.48) is [90]:

$$u(x,t)=U\sum_{j=0}^{\infty}\frac{(-4)^j}{(2j)!}\left(\sum_{m=0}^{j}(-1)^m\binom{j}{m}\frac{e^{-\frac{(x/L)^2}{1+m\lambda^2}}}{\sqrt{1+m\lambda^2}}\right)\left(\frac{tc_0}{L\lambda}\right)^{2j}. \tag{7.51}$$

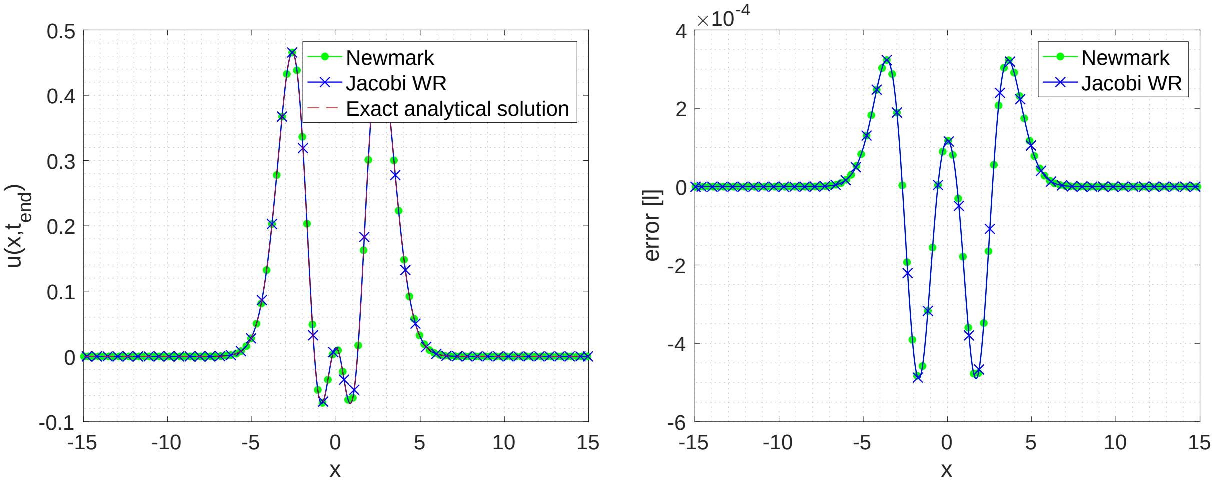
In order to solve the problem in Eq. (7.48) numerically, its infinite domain has to be reduced to a finite size. First, it can be noted that, for given data, one can identify a finite distance δ such that

$$C(x'-x)<10^{-16}, \quad \text{for } |x'-x|\geq\delta, \tag{7.52}$$

meaning that the micromodulus function can be considered to be zero for distances larger than δ. Furthermore, given the nature of the problem and the considered initial conditions, the initial deformation will propagate only for a finite length during $[0,T]$. Hence, we consider a finite domain composed of an internal domain $\Omega=[x_L,x_R]$ and a boundary layer $\mathscr{B}\Omega=[x_L-\delta,x_R+\delta]$ (see Figure), such that for the chosen data,

$$u(x,t)=0\approx u_0(x), \quad \forall(x,t)\in\mathscr{B}\Omega\times[0,T]. \tag{7.53}$$

For our parameters we choose $T=3$ $[s]$, $\lambda=0.75$, $E=1$, $L=1$, $U=1$, $x_L=-15$, $x_R=15$, and $\delta=5$. For time integration a time step $\Delta t=0.005$ $[s]$ was used. The problem is solved by means of an RKPM meshfree approach. The one-dimensional bar was discretized with 1000 nodes and a first order basis, a cubic B-spline kernel and a normalized support size of 1.001 were used. Domain integration was performed by a background mesh with 8 Gauss integration points between every two adjacent nodes. In this work, the transformation method was applied [56][29] and boundary conditions were imposed strongly by collocation.



(a) Displacement at $T = 3$ $[s]$.

(b) Displacement error at $T = 3$ $[s]$.

Figure 7.9: Nonlocal 1D dynamic problem spatially discretized with RKPM: comparison between Newmark and Jacobi WRN_β method. (Left) Comparison of displacement; and (right) corresponding absolute error.

The displacement solution and absolute error obtained by using the WR-Newmark algorithm with Jacobi splitting is presented in Figure 7.9 together with the traditional Newmark solution and the analytical solution. As it can be observed, also for semi-discrete systems arising from meshfree discretizations of nonlocal problems, the proposed WRN_β method are comparable with those obtained from the traditional Newmark scheme.

7.3.3 Two-dimensional plate

This example considers the free vibration of a two-dimensional plate subjected to an initial horizontal displacement. Herein the motion is described by Eq. (7.6) without damping, i.e. $\mathbf{C} = 0$. The problem is solved by means of two different discretization methods: an RKPM meshfree discretization and a a standard finite element discretization. The plate has a total length of $L = 4[m]$ and a height of $H = 1.25[m]$. Initial horizontal displacement with value equal to three times the nodal abscissa coordinate is applied and then immediately released. A fixed boundary displacement condition is present at the leading edge of the plate. Its Young's modulus, density and Poisson's ratio have been chosen to be respectively: $E = 1[\frac{N}{m^2}]$, $\rho = 1[\frac{kg}{m^2}]$ and $\nu = 0.3$. Plane stress conditions are considered. Because of the complexity of the problem, an analytical solution is not available; therefore the solution obtained with the Newmark's method, using a refined time step ($\Delta t = 0.0001[s]$) with respect to the one utilized for the WRN$_\beta$ schemes ($\Delta t = 0.01[s]$), is taken as the reference solution.

RKPM meshfree solution

In this subsection, We repeat the same problem using the RKPM meshfree method. As discussed in Chapter 3, different from finite elements, where the shape functions and the related approximation space are strictly linked to element connectivity, in RKPM the approximate solutions are constructed over a node-based meshfree discretization of the considered domain [56][138] (Figure 7.10), hence the system is denser and not compactly supported.

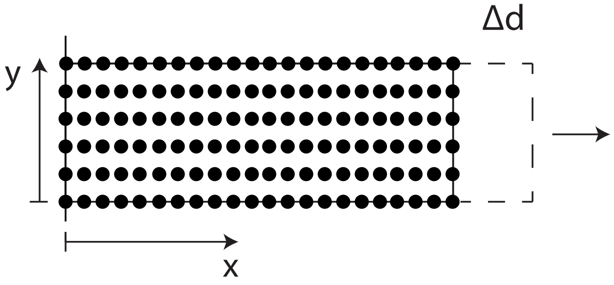
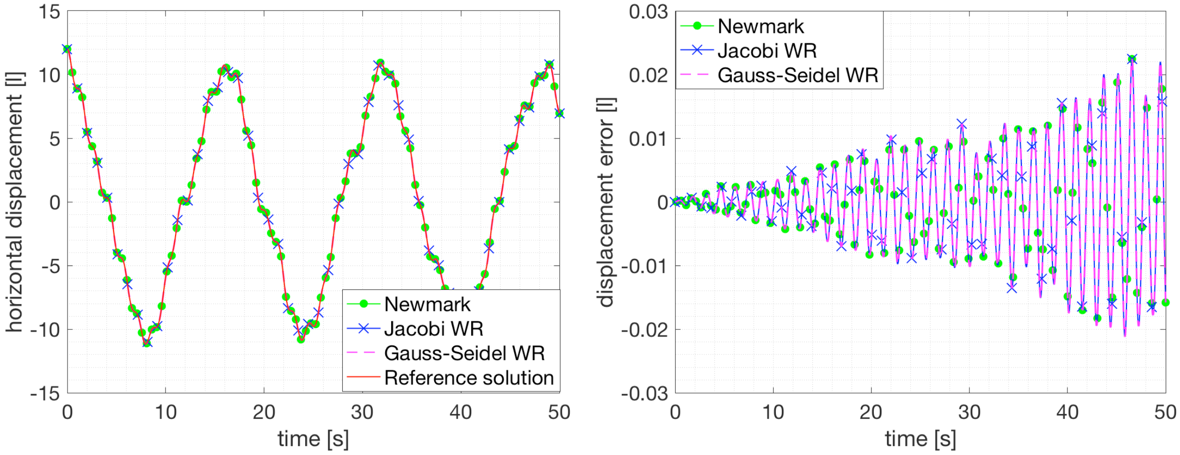


Figure 7.10: Two dimensional plate discretized with discrete points

To assess the accuracy of the WRN$_\beta$ method, the plate was discretized with 54 nodes and a second order basis, a cubic B-spline kernel and a normalized support size of 2.001 in both directions were used. Domain integration was performed by a background mesh with 2x2 Gauss integration over each nodal integration cell. In this work, boundary conditions were imposed through a penalty approach. Figure 7.11(a) shows the horizontal displacement of the free-end midpoint tip computed with the traditional implicit Newmark method and with the point-wise Jacobi and Gauss-Seidel WR-Newmark algorithms, while the corresponding errors of the WRN$_\beta$

schemes are presented in Figure 7.11(b). The L_2 norm values of the absolute error accumulated over time are equal to 0.5651 for all the different considered methods, meaning that the obtained results are comparable.

It has to be noted that the RKPM method recovers the solution and accuracy of a linear finite element solution as the support size approaches 1 if a tent function is chosen as kernel. Given the nonlocal characteristic of the Meshfree approximation, different partitioning choices and integration schemes specifically designed for an RKPM approximation [59][74] might be more appropriate; however, the aim of this example is to show the flexibility of the proposed scheme and its possibility of being applied to systems coming from both Meshfree discretizations and the Finite Element.



(a) Horizontal displacement

(b) Absolute error

Figure 7.11: RKPM meshfree solution; (a) Horizontal displacement of the tip central point over time for the WRN_β methods, and (b) the error compared with a refined Newmark method

The effectiveness of the proposed WRN_β algorithms is shown in Figure 7.12 where the required CPU time for an increasing number of nodes and a time window of one time step was compared to the one required by the traditional Newmark's method. The presented results were obtained for a serial implementation in MATLAB on a macOS machine with a single double-core processor (3.1 GHz Intel Core i7). An LU decomposition was used for the solution step of the standard Newmark method. One can clearly see a trend reversal in terms of CPU time required to solve the problem. Initially, for a small number of nodes, the standard Newmark method performs well and requires the minimum time. However, as the number of nodes increases, the trend is reversed and all WRN_β methods outperform the standard Newmark method, confirming the effectiveness of the Waveform Relaxation Newmark algorithm as a new class of more efficient time integrators.

The main reason is due to the matrix solve time in the Newmark method which increases dramatically with the size of the system, while in the WR Jacobi method there is no need to solve any system (each relaxation iteration is an explicit Newmark scheme) and the WR Gauss-Seidel method requires only to solve forward substitutions. Therefore even if the solution has to be computed as many times as required to obtain convergence of the iterations, its cost is still lower than the one to invert the original dense matrix for complex problems.

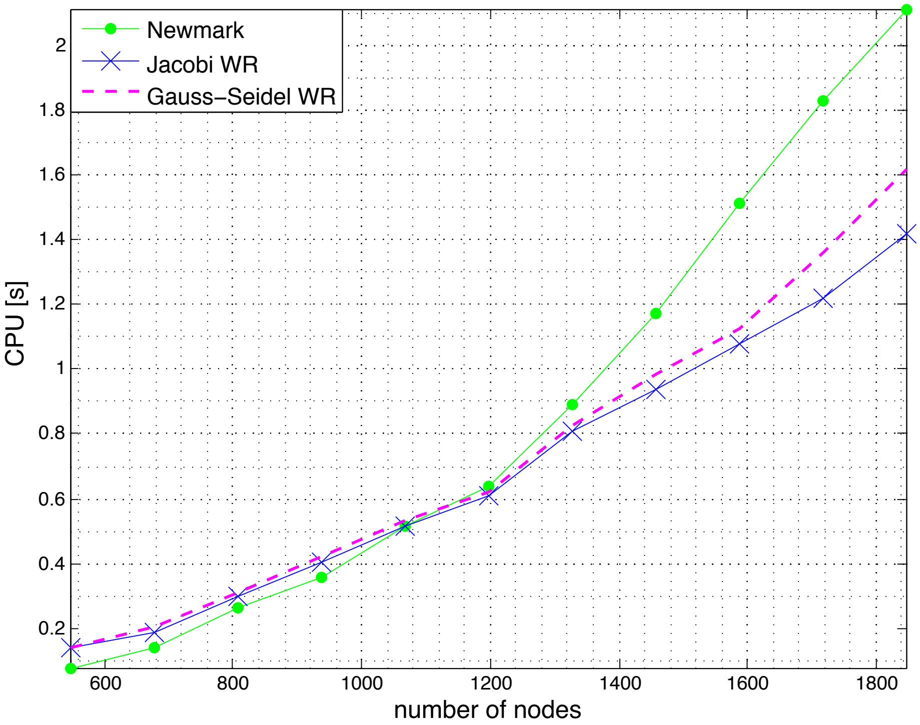


Figure 7.12: CPU time vs number of nodes. Newmark and WRN_β methods. Notice that as the system increases, WRN_β methods converge faster than the Newmark method.

Finite element solution

In this subsection, we repeat the same problem using a finite element discretization (see Figure 7.13).

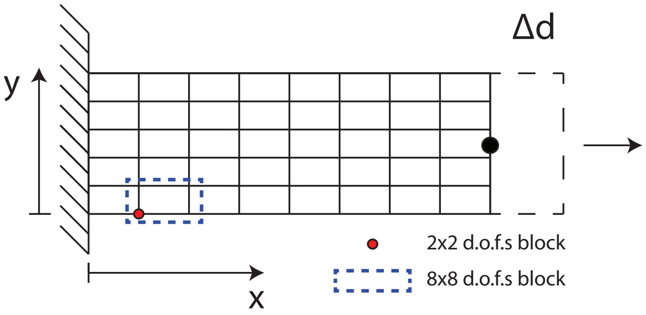
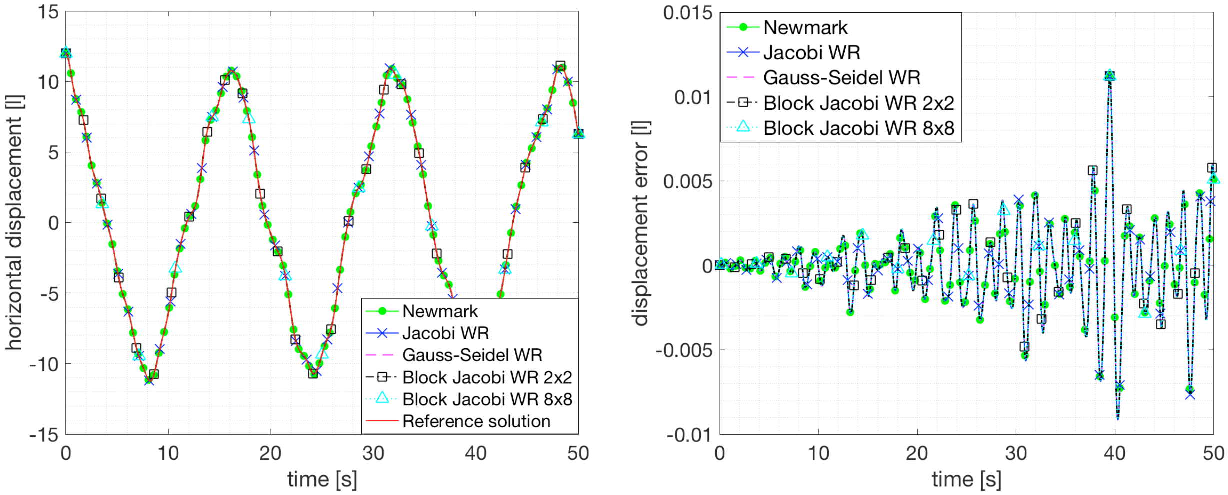


Figure 7.13: Two dimensional plate discretized with finite elements. The nodes on the edge at $x = 0$ are fixed. The degrees of freedom associated with 2x2 and 8x8 partitioning block sizes are shown.

M and **K** are therefore obtained by assembling the mass and stiffness matrices, $\mathbf{m_{el}}$ and $\mathbf{k_{el}}$, respectively, of the bilinear quad elements used for the discretization. The element node numbering has been chosen so as to obtain a largely sparse coefficient matrix. The problem is solved by employing point-wise Jacobi, point-wise Gauss-Seidel and block Jacobi Waveform Relaxation Newmark algorithms. Unless otherwise specified, a point-wise scheme is inferred. In addition, we also consider a block Jacobi WRN_β scheme with blocks containing two and eight degrees of freedom; these, as shown in Figure 7.13, refer to the degrees of freedom associated respectively with single nodes and individual elements.

Figure 7.14(a) shows the displacement of the free-end tip midpoint computed with the standard implicit Newmark method and with the WR-Newmark algorithms for 40 elements, while the corresponding error of the WRN_β schemes is presented in Figure 7.14(b). We note that for all the considered methods the L_2 norm of the absolute error is equal to 0.1743. The absolute errors were computed by comparing the solutions of the different methods to the reference solution over the total time domain. It can be observed that all the WRN_β methods attain the same solution accuracy of the standard Newmark scheme. The maximum natural frequencies for the traditional system used in the Newmark method and for the point-wise Jacobi and Gauss-Seidel ones are presented in Table 7.3. The frequency of the traditional system is greater than those of the partitioned systems.



(a) Horizontal displacement

(b) Absolute error

Figure 7.14: Finite Element solution; (a) Horizontal displacement of the tip central point over time for the WRN_β methods, and (b) the error compared with a refined Newmark method.

Table 7.3: Maximum natural frequency for different methods.

Partitioning scheme	$\omega^h_{max}[Hz]$
standard Newmark	8.36
Jacobi	5.05
Gauss-Seidel	5.05

Figure 7.15 presents the eigenvalues of the error propagation matrix **R** for the point-wise Jacobi-WRN$_\beta$ and the point-wise Gauss-Seidel-WRN$_\beta$ methods for the considered 40 element discretization. Note that the eigenvalues are all within the unit circle and that the Gauss-Seidel-WRN$_\beta$ eigenvalues are more clustered and closer to zero than the Jacobi ones, which indicates

faster convergence of the Gauss-Seidel iterative scheme. Furthermore, Table 7.4 shows the number of iterations and the spectral radius of the error propagation matrix **R** of the point-wise Jacobi and point-wise Gauss-Seidel numerical schemes for different number of elements. The results show that there is almost no change in their spectral radius, and therefore no change in the convergence behaviour in terms of WR iterations, with increase of system size.

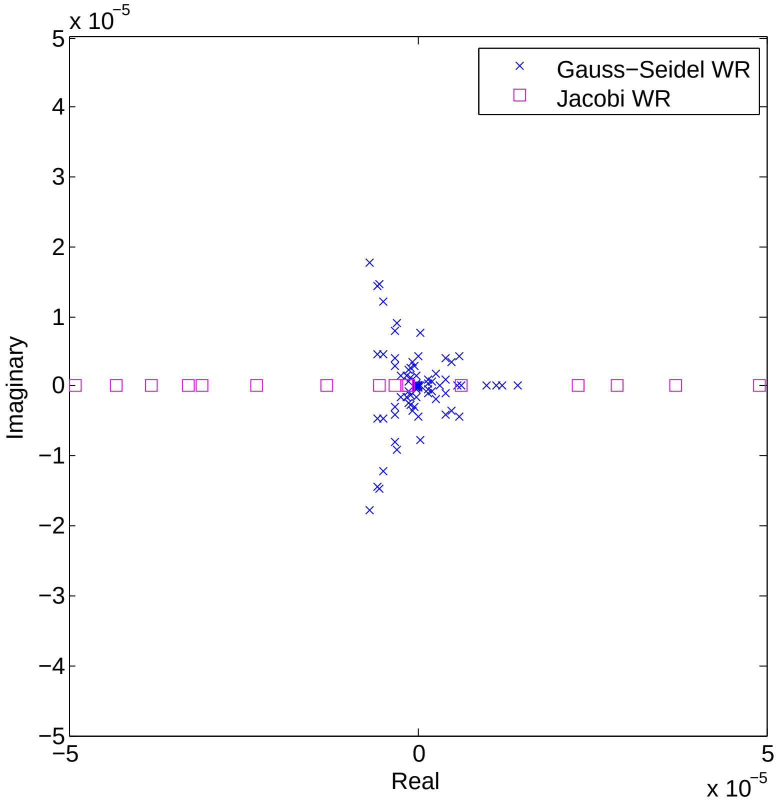


Figure 7.15: Eigenvalues of the error propagation matrix **R** for the 2D plate problem.

Finally, we study the computational efficiency of the proposed WRN$_\beta$ algorithms in the solution of system of linear second order hyperbolic differential equations for an increasing number of elements as compared with the traditional Newmark's method. Figure 7.16 shows the cpu timings for a time window of one time step. The presented results were obtained for a serial implementation in MATLAB on a macOS machine with a single double-core processor

Table 7.4: spectral radii of **R** and number of iterations for increasing number of elements.

-	$\rho(\mathbf{R})$		number of iterations	
number of elements	Jacobi	GS	Jacobi	GS
500	0.5158	0.1223	11	7
1000	0.5310	0.1267	11	7
1500	0.5338	0.1275	11	7
2000	0.5347	0.1277	11	7
2500	0.5350	0.1278	11	7

(3.1 GHz Intel Core i7). An LU decomposition was used for the solution step of the standard Newmark method. Similar to the RKPM results, a clear trend reversal is also observed in the Finite Element time integration, with the WRN_β outperforming the traditional Newmark method as the size of the system increases. This, together with the previous example, confirms that the Waveform Relaxation Newmark algorithm is effective for different structures of the mass and stiffness matrices, regardless whether they arise from a meshfree or a finite element discretization.

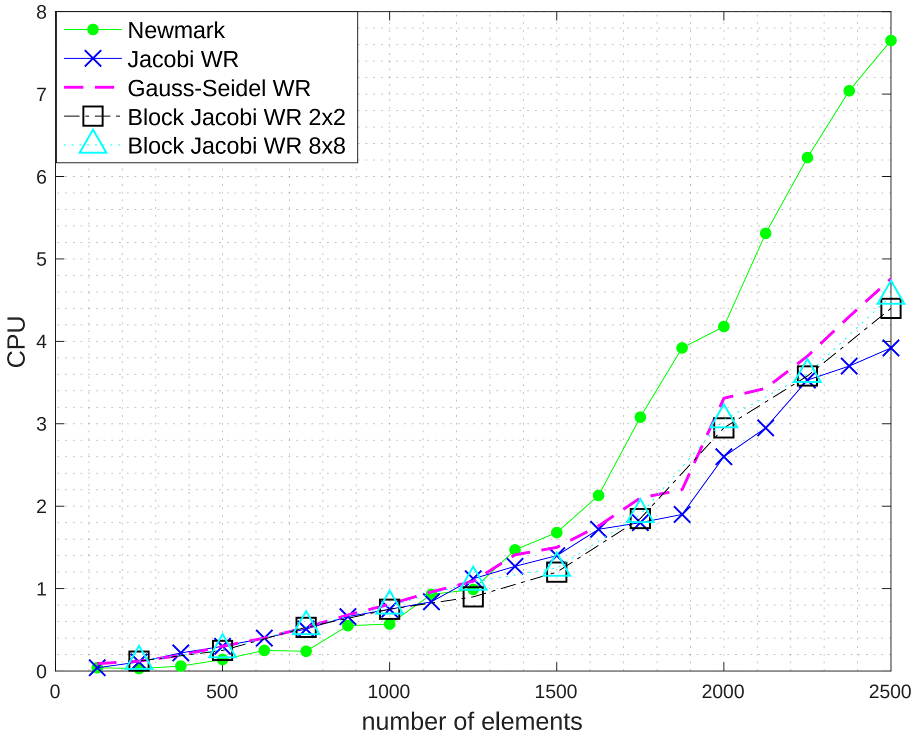


Figure 7.16: CPU time vs number of elements. Newmark and point-wise WRN$_\beta$ methods. Notice that as the system increases, WRN$_\beta$ methods converge faster than the Newmark method.

Acknowledgments

A portion of this chapter has been published in Marco Pasetto, Haim Waisman, J.S. Chen, “A waveform relaxation Newmark method for structural dynamics problems”, *Computational Mechanics*, 63(6), pp. 1223-1242, 2019. The dissertation author was the primary investigator of this material.

7.4 Appendix 7-A

The exact analytical solution to (7.9) can be expressed as:

$$d_1(t) = \hat{X}_{11}[g_{10}\cos(\omega_1 t) + \frac{g_{\dot{1}0}}{\omega_1}\sin(\omega_1 t)] + \hat{X}_{12}[g_{20}\cos(\omega_2 t) + \frac{g_{\dot{2}0}}{\omega_2}\sin(\omega_2 t)]$$

$$d_2(t) = \hat{X}_{21}[g_{10}\cos(\omega_1 t) + \frac{g_{\dot{1}0}}{\omega_1}\sin(\omega_1 t)] + \hat{X}_{22}[g_{20}\cos(\omega_2 t) + \frac{g_{\dot{2}0}}{\omega_2}\sin(\omega_2 t)]$$

Where the analytical expressions for $\hat{X}_{11}, \hat{X}_{12}, \hat{X}_{21}, \hat{X}_{22}, g_{10}, g_{\dot{1}0}, g_{20}, g_{\dot{2}0}, \omega_1$ and ω_2 are as follows:

$$\hat{X}_{11} = \frac{k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a}}{2k_2 m_1 \sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$\hat{X}_{12} = \frac{k_1 m_1 - k_1 m_2 - k_2 m_2 - \hat{a}}{2k_2 m_1 \sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$\hat{X}_{21} = \frac{1}{\sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$\hat{X}_{22} = \frac{1}{\sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$g_{10} = \frac{d_{1,0}\hat{a} + d_{1,0}k_1 m_1 - d_{1,0}k_1 m_2 - d_{1,0}k_2 m_2 + 2d_{2,0}k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$g_{20} = \frac{-d_{1,0}\hat{a} + d_{1,0}k_1 m_1 - d_{1,0}k_1 m_2 - d_{1,0}k_2 m_2 + 2d_{2,0}k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}}$$

$$\dot{g_{10}} = \frac{v_{1,0}\hat{a} + v_{1,0}k_1m_1 - v_{1,0}k_1m_2 - v_{1,0}k_2m_2 + 2v_{2,0}k_2m_2}{2k_2\sqrt{m_2 + \frac{(k_1m_1 - k_1m_2 - k_2m_2 + \hat{a})^2}{4k_2^2m_1}}}$$

$$\dot{g_{20}} = \frac{-v_{1,0}\hat{a} + v_{1,0}k_1m_1 - v_{1,0}k_1m_2 - v_{1,0}k_2m_2 + 2v_{2,0}k_2m_2}{2k_2\sqrt{m_2 + \frac{(-k_1m_1 + k_1m_2 + k_2m_2 + \hat{a})^2}{4k_2^2m_1}}}$$

$$\omega_1 = \sqrt{\frac{k_1m_1 + k_1m_2 + k_2m_2 - \hat{a}}{2m_1m_2}}$$

$$\omega_2 = \sqrt{\frac{k_1m_1 + k_1m_2 + k_2m_2 + \hat{a}}{2m_1m_2}}$$

where

$$\hat{a} = \sqrt{k_1^2m_1^2 - 2k_1^2m_1m_2 + k_1^2m_2^2 - 2k_1k_2m_1m_2 + 2k_1k_2m_2^2 + 4k_2^2m_1m_2 + k_2^2m_2^2}$$

The results shown in Figure 7.4a and Figure 7.4b were obtained for $m_1 = 6[kg]$, $m_2 = 3[kg]$, $k_1 = \frac{1}{6}[\frac{N}{m}]$, $k_2 = \frac{1}{6}[\frac{N}{m}]$, $d_{1,0} = 6[m]$, $d_{2,0} = 12[m]$, $v_{1,0} = 0[\frac{m}{s}]$ and $v_{2,0} = 0[\frac{m}{s}]$.

7.5 Appendix 7-B

Initialization ν=0:

$$\begin{cases} d_1^{(0)}(t) = 6 \\ d_2^{(0)}(t) = 12 \end{cases}$$

Jacobi WR:

- Iteration ν=1:

$$\begin{cases} 6\dot{d}_1^{(1)}(t) + \frac{1}{3} d_1^{(1)}(t) = 2 \\ 3\dot{d}_2^{(1)}(t) + \frac{1}{6} d_2^{(1)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(1)}(t) = 6 \\ d_2^{(1)}(t) = 6[1 + \cos(\frac{t}{3\sqrt{2}})] \end{cases}$$

- Iteration ν=2:

$$\begin{cases} 6\dot{d}_1^{(2)}(t) + \frac{1}{3} d_1^{(2)}(t) = 1 + \cos(\frac{t}{3\sqrt{2}}) \\ 3\dot{d}_2^{(2)}(t) + \frac{1}{6} d_2^{(2)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(2)}(t) = 3 + 3\cos(\frac{t}{3\sqrt{2}}) + \frac{t}{2\sqrt{2}}\sin(\frac{t}{3\sqrt{2}}) \\ d_2^{(2)}(t) = 6[1 + \cos(\frac{t}{3\sqrt{2}})] \end{cases}$$

- Iteration ν=3:

$$\begin{cases} 6\ddot{d}_1^{(3)}(t) + \frac{1}{3}d_3^{(2)}(t) = 1 + \cos(\frac{t}{3\sqrt{2}}) \\ 3\ddot{d}_2^{(3)}(t) + \frac{1}{6}d_3^{(2)}(t) = \frac{1}{2} + \frac{1}{2}\cos(\frac{t}{3\sqrt{2}}) + \frac{t}{12\sqrt{2}}\sin(\frac{t}{3\sqrt{2}}) \end{cases}$$

Solution:

$$\begin{cases} d_1^{(3)}(t) = 3 + 3\cos(\frac{t}{3\sqrt{2}}) + \frac{t}{2\sqrt{2}}\sin(\frac{t}{3\sqrt{2}}) \\ d_2^{(3)}(t) = 3 + (9 - \frac{t^2}{48})\cos(\frac{t}{3\sqrt{2}}) + \frac{5t}{8\sqrt{2}}\sin(\frac{t}{3\sqrt{2}}) \end{cases}$$

- ...

<u>Gauss-Seidel WR</u>:

- Iteration ν=1:

$$\begin{cases} 6\ddot{d}_1^{(1)}(t) + \frac{1}{3}d_1^{(1)}(t) = 2 \\ 3\ddot{d}_2^{(1)}(t) + \frac{1}{6}d_2^{(1)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(1)}(t) = 6 \\ d_2^{(1)}(t) = 6[1 + \cos(\dfrac{t}{3\sqrt{2}})] \end{cases}$$

• Iteration ν=2:

$$\begin{cases} 6\dot{d}_1^{(2)}(t) + \dfrac{1}{3}d_1^{(2)}(t) = 1 + \cos(\dfrac{t}{3\sqrt{2}}) \\ 3\dot{d}_2^{(2)}(t) + \dfrac{1}{6}d_2^{(2)}(t) = \dfrac{1}{2} + \dfrac{1}{2}\cos(\dfrac{t}{3\sqrt{2}}) + \dfrac{t}{12\sqrt{2}}\sin(\dfrac{t}{3\sqrt{2}}) \end{cases}$$

Solution:

$$\begin{cases} d_1^{(2)}(t) = 3 + 3\cos(\dfrac{t}{3\sqrt{2}}) + \dfrac{t}{2\sqrt{2}}\sin(\dfrac{t}{3\sqrt{2}}) \\ d_2^{(2)}(t) = 3 + (9 - \dfrac{t^2}{48})\cos(\dfrac{t}{3\sqrt{2}}) + \dfrac{5t}{8\sqrt{2}}\sin(\dfrac{t}{3\sqrt{2}}) \end{cases}$$

• Iteration ν=3:

$$\begin{cases} 6\dot{d}_1^{(3)}(t) + \dfrac{1}{3}d_3^{(2)}(t) = \dfrac{1}{6}[3 + (9 - \dfrac{t^2}{48})\cos(\dfrac{t}{3\sqrt{2}}) + \dfrac{5t}{8\sqrt{2}}\sin(\dfrac{t}{3\sqrt{2}})] \\ 3\dot{d}_2^{(3)}(t) + \dfrac{1}{6}d_3^{(2)}(t) = \dfrac{1}{6}[\dfrac{3}{2} - \dfrac{1}{64}(-288 + t^2)\cos(\dfrac{t}{3\sqrt{2}}) + \\ \qquad\qquad\qquad\qquad - \dfrac{\sqrt{2}t}{3456}(-1458 + t^2)\sin(\dfrac{t}{3\sqrt{2}})] \end{cases}$$

Solution:

$$\begin{cases} d_1^{(3)}(t) = \dfrac{3}{2} - \dfrac{1}{64}(-288+t^2)\cos(\dfrac{t}{3\sqrt{2}}) - \dfrac{\sqrt{2}t}{3456}(-1458+t^2)\sin(\dfrac{t}{3\sqrt{2}}) \\ d_2^{(3)}(t) = \dfrac{3}{2} + \dfrac{1}{82944}(870912-3294t^2+t^4)\cos(\dfrac{t}{3\sqrt{2}}) + \\ \qquad\qquad + \dfrac{\sqrt{2}t}{13824}(6831-7t^2)\sin(\dfrac{t}{3\sqrt{2}}) \end{cases}$$

- ...

7.6 Appendix 7-C

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{K}\mathbf{d}_{n+1} = \mathbf{f}_{n+1}$$

$$(\mathbf{M}_+ - \mathbf{M}_-)\ddot{\mathbf{d}}_{n+1} + (\mathbf{K}_+ - \mathbf{K}_-)\mathbf{d}_{n+1} = \mathbf{f}_{n+1}$$

$$\mathbf{M}_+\ddot{\mathbf{d}}^{\nu+1}_{n+1} + \mathbf{K}_+\mathbf{d}^{\nu+1}_{n+1} = \mathbf{f}_{n+1} + \mathbf{M}_-\ddot{\mathbf{d}}^{\nu}_{n+1} + \mathbf{K}_-\mathbf{d}^{\nu}_{n+1}$$

Now considering the Newmark's scheme with $\Delta t \neq 0$ and predictor:

$$\hat{\mathbf{a}}^{\nu+1}_{n+1} = \frac{1}{\beta\Delta t^2}\mathbf{d}^{\nu+1}_n + \frac{1}{\beta\Delta t}\dot{\mathbf{d}}^{\nu+1}_n + (\frac{1}{2\beta} - 1)\ddot{\mathbf{d}}^{\nu+1}_n$$

The solution phase then becomes:

$$(\frac{\mathbf{M}_+}{\beta\Delta t^2} + \mathbf{K}_+)\mathbf{d}^{\nu+1}_{n+1} = \mathbf{f}_{n+1} + \mathbf{K}_-\mathbf{d}^{\nu}_{n+1} + \mathbf{M}_-\ddot{\mathbf{d}}^{\nu}_{n+1} - \mathbf{M}_+\hat{\mathbf{a}}^{\nu+1}_{n+1}$$

From the corrector step coming from the previous iteration ν, $\ddot{\mathbf{d}}^{\nu}_{n+1} = \frac{1}{\beta\Delta t^2}\mathbf{d}^{\nu}_{n+1} - \hat{\mathbf{a}}^{\nu}_{n+1}$, leading to:

$$(\frac{\mathbf{M}_+}{\beta\Delta t^2} + \mathbf{K}_+)\mathbf{d}^{\nu+1}_{n+1} = \mathbf{f}_{n+1} + (\mathbf{K}_- + \frac{\mathbf{M}_-}{\beta\Delta t^2})\mathbf{d}^{\nu}_{n+1} - \mathbf{M}_-\hat{\mathbf{a}}^{\nu}_{n+1} - \mathbf{M}_+\hat{\mathbf{a}}^{\nu+1}_{n+1}$$

Rewrite as:

$$\mathbf{P}\mathbf{d}^{\nu+1} = \mathbf{Q}\mathbf{d}^{\nu} + \tilde{\mathbf{f}}$$

where the subscript $(n+1)$ has been dropped and where $\mathbf{P} = \frac{\mathbf{M}_+}{\beta\Delta t^2} + \mathbf{K}_+$, $\mathbf{Q} = \mathbf{K}_- + \frac{\mathbf{M}_-}{\beta\Delta t^2}$ and $\tilde{\mathbf{f}} = \mathbf{f} - \mathbf{M}_-\hat{\mathbf{a}}^{\nu} - \mathbf{M}_+\hat{\mathbf{a}}^{\nu+1}$, which leads to

$$\mathbf{d}^{\nu+1} = \mathbf{P}^{-1}\mathbf{Q}\mathbf{d}^{\nu} + \mathbf{P}^{-1}\tilde{\mathbf{f}}$$

The relationship between the error of two successive iterations can be written as:

$$\mathbf{e}^{\nu+1} = \mathbf{P}^{-1}\mathbf{Q}\mathbf{e}^{\nu} = \mathbf{R}\mathbf{e}^{\nu}$$

where $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$ is the error propagation matrix. For convergence:

$$\rho(\mathbf{R}) < 1$$

$$\rho(\mathbf{P}^{-1}\mathbf{Q}) < 1$$

$$\rho[(\frac{\mathbf{M}_+}{\beta\Delta t^2} + \mathbf{K}_+)^{-1}(\mathbf{K}_- + \frac{\mathbf{M}_-}{\beta\Delta t^2})] < 1$$

$$\rho[(\mathbf{M}_+ + \beta\Delta t^2\mathbf{K}_+)^{-1}(\mathbf{M}_- + \beta\Delta t^2\mathbf{K}_-)] < 1$$

Chapter 8

Conclusions

8.1 Conclusions

Meshfree discretizations of local and nonlocal mechanics (e.g., peridynamics) have been shown to be effective for problems involving large deformations, crack propagations or material fragmentation. However, achieving good accuracy while keeping a low computational cost in such numerical simulations can be difficult.

To this end, for local problems, a Lagrangian/semi-Lagrangian reproducing kernel coupling approach was developed in a meshfree framework by blending the shape functions and shape function gradients of the Lagrangian RK and semi-Lagrangian RK approximations. This method allows for the use of semi-Lagrangian RK approximation only in those portions of the domain where it is required by the nature of the deformation, i.e. the deformation gradient is no longer invertible and the Lagrangian RK formulation breaks down. In this way, large deformation and fragmentation problems can be solved at a reduced computational cost with respect to full semi-Lagrangian simulations. The continuity and stability of the proposed coupling approach were analysed and its ability to save computational run-time compared to full semi-Lagrangian simulations was demonstrated through a suite of numerical examples.

For nonlocal problems, we proposed an RK enhanced approach for strong form peridynamics. The introduction of RK shape functions in the solution approximation allows for arbitrary smoothness and completeness of the approximation. This allowed for the achievement of higher convergence rates with respect to the commonly used node-based approach, which is based on a piecewise-constant approximation of numerical solutions is characterized by a convergence rate limited to first order. Another key element in achieving higher convergence rates was the integration scheme. Inaccurate integration can lead to oscillatory convergence behavior. To mitigate this behavior, a Gauss integration scheme that positioned Gauss integration points into a Cartesian background grid was considered. The ability of the RK enhanced peridynamic approach to achieve high-order convergence was demonstrated through the solution of 1D and

2D bond-based static linear peridynamic problems characterized by quadratic, cubic (1D and 2D), and exponential (1D) displacement fields. A preliminary study for problems involving cracks was also performed.

Also, for the variational form of peridynamics *n*-th order integration constraints were identified. This was achieved by first considering an arbitrary order polynomial solution field and associated body force and boundary conditions. The RK approximation (which can be easily built to be *n*-th order complete) was then employed for the trial and test functions. By introducing numerical integration in the weak form, we obtained the conditions (i.e., integration constraints) that the discrete variational form of peridynamics needs to satisfy in order to be *n*-th order variationally consistent. This lead to the construction of a high-order symmetrical nodal quadrature scheme, with the goal of reducing the costs associated with the integration procedure required for the solution of variational form peridynamic problems. To this end, we first considered a nodal discretization of the peridynamic weak form, which turned the double integration present in the variational equation into a double-sum. However, instead of considering each integration weight for each node separately, we considered a single integration weight for each couple of nodes present in the double summation, so to retain their symmetry of interaction at the discrete level. Following an idea similar to the one employed to construct the RK shape functions, we then proceeded to define these integration weights as the combination of a chosen basis and a set of unknown coefficients to be determined through the imposition of the *n*-th order integration constraints. As long as the resulting system can be solved and the unknown coefficients determined, the resulting integration weights will be symmetrical and satisfy *n*-th order variational consistency. This was verified through the solution a one-dimensional variational form static peridynamic example, discretized using both a uniform and non-uniform meshfree grid.

Finally, to further reduce the overall computational cost, a Waveform Relaxation Newmark algorithm for the solution of linear structural dynamics hyperbolic systems was developed by pairing the implicit form of the Newmark's method and the iterative waveform relaxation schemes. This method has several notable features as compared with traditional Newmark methods: it requires iterations in the time domain, however, the convergence can be obtained quickly depending on the integration time window size, it retains unconditional stability but requires significantly lower computational costs which is emphasized for larger problem sizes, it is unstructured in time domain and well suited for time parallelization compared with the inherently sequential Newmark method.

8.2 Recommendations for future work

The work presented in this dissertation, in the author's opinion, shows some potential for further investigations. Some suggested directions for future research are the following:

1) For the proposed blended Lagrangian/semi-Lagrangian RK approach:

- Investigation of the effect of different ramping functions (for example smoother ones) from the linear one here employed.
- Study of the completeness of the blended approach in the current configuration.
- Numerical verification of the theoretical time stability analysis presented in this work.
- Incorporation of other integration schemes, such as the variationally consistent integration (VCI) and the naturally stabilized nodal integration (NSNI) in the blended approach.

2) For the proposed RK Enhanced Peridynamic approach:

- Further investigation of the proposed RK Enhanced Peridynamic approach to problems involving dynamics, cracks and discontinuities. Though here a preliminary study using Lagrangian RK shape functions was conducted using nodal integration, the effect of employing higher order quadrature in conjunction with RK for these problems is still to be studied.

Furthermore, the use of Lagrangian RK shape functions can lead to interaction between particles that have had their bond broken due to the reconstruction of the displacement field. The use of a reconstructed approximation (i.e., of semi-Lagrangian RK) could be investigated and maybe used in conjunction with the blended Lagrangian/semi-Lagrangian approach, in order to reduce computational cost.

- Study of the effect of field approximation for problems involving cracks. When a field approximation or interpolation is used, the displacement at a point near the boundary of a neighbourhood can happen to be evaluated using information from outside the neighbourhood itself: this effectively increases the size of the nonlocal interaction. While this was shown not to majorly affect convergence behaviour for smooth problems by means of studying the effect of changing the RK approximation support size, this could be investigated also for problems involving discontinuities.

- Extension of the work to nonlinear problems.

3) For the variationally consistent integration in weak form peridynamics:

- Test the proposed approach in higher dimensions.

- Develop and implement a Nitsche's approach for the nonlocal peridynamic equation so to impose Dirichlet's boundary conditions without zeroing out the test functions on the Dirichlet's boundary.

- Check the performance of the proposed approach in improving convergence behaviour of peridynamic meshfree numerical solutions. This could be achieved by looking at the resulting convergence rates for problems with non-polynomial solutions.

- Compare the integration weights obtained through the proposed variationally consistent integration with those obtained with other quadrature rules, to see if any similarities are present.

4) For the waveform relaxation Newmark method:

- Application of the scheme to general nonlinear structural problems
- Implement the method in parallel.
- Study optimal splitting schemes for denser systems like the one coming from RKPM and Isogeometric discretizations.

Acknowledgments

A portion of this chapter has been published in Marco Pasetto, Haim Waisman, J.S. Chen, "A waveform relaxation Newmark method for structural dynamics problems", *Computational Mechanics*, 63(6), pp. 1223-1242, 2019. The dissertation author was the primary investigator of this material.

A portion of this chapter has been published in M. Pasetto, Y. Leng, J.S. Chen, J.T. Foster, P. Seleson, "A reproducing kernel enhanced approach for peridynamic solutions", *Computer Methods in Applied Mechanics and Engineering*, 340, 2018. The dissertation author was the primary investigator of the included material.

A portion of this chapter is currently being prepared for submission for publication of the material. M. Pasetto, J. Baek, J.S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, "A Lagrangian/semi-Lagrangian Coupling Approach for Meshfree Modeling of Extremely Large Deformation Problems". The dissertation author was the primary investigator of this material.

A portion of this chapter is currently being prepared for submission for publication of the material. M. Pasetto, J.S. Chen, "Variationally Consistent Integration in Reproducing Kernel Enhanced Weak Form Peridynamics". The dissertation author was the primary investigator of this material.

Bibliography

[1] A. Huerta, S. Fernandez-Mendez. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48:1615–1636, 2000.

[2] M. Zennaro A.Bellen. Parallel algorithms for initial-value problems for difference and differential equations. *Journal of Computational and Applied Mathematics*, 25:341–350, 1989.

[3] A.J. Christlieb, C.B. Macdonald, B.W. Ong. Parallel high-order integrators. *SIAM Journal on Scientific Computing*, 32(2):818–835, 2010.

[4] B. Aksoylu, M.L. Parks. Variational theory and domain decomposition for nonlocal problems. *Applied Mechanics and Computation*, 217:6498–6515, 2011.

[5] B. Alali, K. Liu, M. Gunzburger. A Generalized Nonlocal Calculus with Application to the Peridynamics Model for Solid Mechanics. *arXiv e-prints*, February 2014.

[6] B. Kilic, E. Madenci. Prediction of crack paths in a quenched glass plate by using peridynamic theory. *International Journal of Fracture*, 156:165–177, 2009.

[7] B. Ren, C.T. Wu, E. Askari. A 3D discontinuous Galerkin finite element method with the bond-based peridynamics model for dynamic brittle failure analysis. *11th European LS-DYNA Conference 2017, Salzburg, Austria*, 2017.

[8] S. Bond. Quadrature for nonlocal mechanics and peridynamics. Presentation at the 12th U.S. National Congress on Computational Mechanics, July 22–25, 2013, Raleigh, NC, USA.

[9] C. Diyaroglu, E. Oterkus, E. Madenci, T. Rabczuk, A. Siddiq. Peridynamic modeling of composite laminates under explosive loading. *Composite Structures*, 144:14–23, 2016.

[10] D. Hegen. Element-free Galerkin mehods in combination with finite element approaches. *Computer Methods in Applied Mechanics and Engineering*, 135:143–166, 1996.

[11] D. Sheen, I.H. Sloan, V. Thome. A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature. *Mathematics of Computation*, 69(299):177–195, 1999.

[12] David Littlewood. Roadmap for software implementation. In F. Bobaru, J.T. Foster, P.H. Geubelle and S.A. Silling, editor, *Handbook of Peridynamic Modeling*. CRC Press, 2017.

[13] D.Huang, Q. Zhang, P. Qiao. Damage and progressive failure of concrete structures using non-local peridynamic modeling. *Science China Technological Sciences*, 54(3):591–596, 2011.

[14] D.J. Littlewood. A nonlocal approach to modeling crack nucleation in AA 7075-T651. *Proceedings of ASME 2011 International Mechanical Engineering Congress and Exposition*, 2011.

[15] E. Askari, F. Bobaru, R.B. Lehoucq, M.L. Parks, S.A. Silling, O. Weckner. Peridynamics for multiscale materials modeling. *Journal of Physics: Conference Series*, 125, 2008.

[16] E. Emmrich, O. Weckner. The peridynamic equation and its spatial discretization. *Mathematical Modelling and Analysis*, 12(1):17–27, 2007.

[17] E. Lelarasmee, A.E. Ruehli, A.L. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 1(3), 1982.

[18] E. Madenci, E. Oterkus. *Peridynamic Theory and its Applications*. Springer, 2013.

[19] E. Oterkus, E. Madenci. Peridynamic analysis of fiber-reinforced composite materials. *Journal of Mechanics of Materials and Structures*, 7(1):45–84, 2012.

[20] E. Yreux, J.S. Chen. A quasi-linear reproducing kernel particle method. *International Journal for Numerical Methods in Engineering*, 109:1045–1064, 2017.

[21] E.M. Williams, S.S. Graham, P.A. Reed, T.S. Rushing. Laboratory Characterization of Cor-Tuf Concrete With and Without Steel Fibers. *Tech. rep ERDC/GSL*, TR-09-22, 2009.

[22] Erdogan Madenci, Mehmet Dorduncu, Atila Barut, Nam Phan. Weak form of peridynamics for nonlocal essential and natural boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 337:598–631, 2018.

[23] F. Bobaru, G. Zhang. Why do cracks branch? A peridynamic investigation of dynamic brittle fracture. *International Journal of Fracture*, 196:59–98, 2015.

[24] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling. *Handbook of Peridynamic Modeling*. CRC Press, 2017.

[25] F. Bobaru, M. Yang, L.F. Alves, S.A. Silling, E. Askari, J. Xu. Convergence, adaptive refinement, and scaling in 1D peridynamics. *International Journal for Numerical Methods in Engineering*, 77:852–877, 2009.

[26] F. Bobaru, Y.D. Ha. Adaptive refinement and multi scale modeling in 2d peridynamics. *Journal for Multiscale Computational Engineering*, 9:635–659, 2011.

[27] F. Xu, M. Gunzburger, J. Burkardt. A multiscale method for nonlocal mechanics and diffusion and for the approximation of discontinuous functions. *Computer Methods in Applied Mechanics and Engineering*, 307:117–143, 2016.

[28] F. Xu, M. Gunzburger, J. Burkardt, Q. Du. A multiscale implementation based on adaptive mesh refinement for the nonlocal peridynamics model in one dimension. *Multiscale Modeling & Simulation*, 14(1):398–429, 2016.

[29] F.C. Günter, W.K. Liu. Implementation of boundary conditions for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 163:205–230, 1998.

[30] F.P. Bowden, J.H. Brunton, J.E. Field, A.D. Heyes. Controlled Fracture of Brittle Solids and Interruptions of Electrical Current. *Nature*, 216:38–42, 2005.

[31] S. Vandewalle G. Horton. A space-time multigrid method for parabolic partial differential equations. *SIAM Journal on Scientific Computing*, 16:848–864, 1995.

[32] G. Horton, S. Vandewalle. A space-time multigrid method for parabolic partial differential equations. *SIAM Journal on Scientific Computing*, 16(4):848–864, 1995.

[33] T. J. R. Hughes G. M. Hulbert. Space-time finite element methods for second order hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 84:327–348, 1990.

[34] G. Taylor. The use of flat-ended projectiles for determining dynamic yield stress, I. Theoretical considerations. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 194(1038):289–299, 1948.

[35] M. J. Gander. 50 years of time parallel time integration. *Contributions in Mathematical and Computational Sciences*, 9:69–113, 2015.

[36] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.

[37] G.C. Ganzenmüller, S. Hiermaier, M. May. On the similarity of meshless discretizations of Peridynamics and Smooth-Particle Hydrodynamics. *Computers and Structures*, 150:71–78, 2015.

[38] G.R. Johnson. Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations. *Nuclear Engineering and Design*, 150:256–274, 1994.

[39] G.R. Johnson, R.A. Stryk, S.R. Beissel. SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering*, 139:247–373, 1996.

[40] P.-C. Guan. *Adaptive Coupling of FEM and RKPM Formulations for Contact and Impact Problems*. PhD thesis, University of California Los Angeles, 2009.

[41] H. Waisman, J. Fish. A space-time multilevel method for molecular dynamics simulations. *Computer Methods in Applied Mechanics and Engineering*, 195 (44-47):6542–6559, 1996.

[42] H. Wang, H. Tiao. A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model. *Journal of Computational Physics*, 231:7730–7738, 2012.

[43] J.S. Chen H. Wei. A damage particle method for smeared modeling of brittle fracture. *International Journal for Multiscale Computational Engineering*, 16(4):303–324, 2018.

[44] H. Wei, J.S. Chen, M. Hillman. A stabilized nodally integrated meshfree formulation for fully coupled hydro-mechanical analysis of fluid-saturated porous media. *Computers & Fluids*, 141:105–115, 2016.

[45] T. J. R. Hughes. *The Finite Element Method – Linear Static and Dynamic Finite Element Analysis*. Courier Dover Publications, Spring 2012.

[46] J.S. Chen J. J. Koester. Conforming window functions for meshfree methods. *Computer Methods in Applied Mechanics and Engineering*, 347:588–621, 2019.

[47] S. Vandewalle J. Janssen. Multigrid waveform relaxation on spatial finite element meshes: the continuous-time case. *SIAM Journal on Numerical Analysis*, 33 (2):456–474, 1996.

[48] S. Vandewalle J. Janssen. Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM Journal on Scientific Computing*, 17 (1):133–155, 1996.

[49] S. Vandewalle J. Janssen. On sor waveform relaxation methods. *SIAM Journal on Numerical Analysis*, 34:2456–2481, 1997.

[50] Y.-L. Jiang" "J. Liu. A parareal algorithm based on waveform relaxation. *Mathematics and Computers in Simulation*, 82:2167–2181, 2012.

[51] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733, 1964.

[52] J. White, F. Odeh, A. Sangiovanni-Vincentelli, A. Ruehli. Waveform relaxation: Theory and practice. *Transactions of The Society for Computer Simulation*, 2:95–133, 1985.

[53] J.A. Mitchell, S.A. Silling, D.J. Littlewood. A position-aware linear solid constitutive model for peridynamics. *Journal of Mechanics of Materials and Structures*, 10 (5):539–557, 2015.

[54] J.J Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[55] Y. Wu J.S. Chen. Stability in Lagrangian and Semi-Lagrangian Reproducing Kernel Discretizations Using Nodal Integration in Nonlinear Solid Mechanics. In *Leito V.M.A., Alves C.J.S., Armando Duarte C. (eds) Advances in Meshfree Techniques. Computational Methods in Applied Sciences*, volume 5, pages 55–76. Springer, 2007.

[56] J.S. Chen, C. Pan, C.-T. Wu, W.K. Liu. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer methods in Applied Mechanics and Engineering*, 139:195–227, 1996.

[57] J.S. Chen, C. Pan, C.M.O.L Roque, H.P. Wang. A Lagrangian reproducing kernel particle methods for metal forming analysis. *Computational Mechanics*, 22:289–307, 1998.

[58] J.S. Chen, C. Pan, C.T. Wu. Large deformation analysis of rubber based on reproducing kernel particle method. *Computational Mechanics*, 19(3):211–227, 1997.

[59] J.S. Chen, C.-T. Wu, S. Yoon, Y. You. A Stabilized Conformal Nodal Integration for Galerkin Mesh-free Methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[60] J.S. Chen, H.-P. Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer methods in applied mechanics and engineering*, 187:441–468, 2000.

[61] J.S. Chen, M. Hillman, M. Rüter. An Arbitrary Order Variationally Consistent Integration for Galerkin Meshfree Methods. *International Journal for Numerical Methods in Engineering*, 95:387–418, 2013.

[62] J.S. Chen, M. Hillman, S.-W. Chi. Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics*, 143(4), 2017.

[63] J.S. Chen, S.W. Chi, C.H. Lee, C. Marodon, M.J. Roth, T.R. Slawson. A Multi Scale Meshfree Approach for Modeling Fragment Penetration into Ultra-Strength Concrete. *Tech. rep ERDC/GSL*, TR-11-35, 2011.

[64] J.S. Chen, W. Han, Y. You, X. Meng. A reproducing kernel method with nodal interpolation property. *International Journal of Numerical Methods in Engineering*, 56(7):935–960, 2003.

[65] J.S. Chen, W. Hu, M.A. Puso, Y. Wu, X. Zhang. Strain smoothing for stabilization and regularization of Galerkin meshfree methods. In M. Griebel and M.A. Schweitzer, editor, *Meshfree Methods for Partial Differential Equations III*, pages 57–75. Springer, Berlin, 2007.

[66] J.S. Chen, X. Zhang, T. Belytschko. An implicit gradient model by a reproducing kernel strain regularization in strain localization problems. *Computer methods in applied mechanics and engineering*, 193:2827–2844, 2003.

[67] S. Yamada K. Osawa. Waveform relaxation for second order differential equation y”=f(x,y). *Lecture Notes in Computer Science*, 1470:780–787, 1998.

[68] E. Lelarasmee. *The Waveform Relaxation Method for Time Domain Analysis of Large Scale Integrated Circuits: Theory and Applications*. PhD thesis, EECS Department, University of California, Berkeley, 1982.

[69] M. D’Elia, Q. Du, M. Gunzburger. Recent progress in mathematical and computational aspects of peridynamics. In G. Voyiadjis, editor, *Handbook of Nonlocal Continuum Mechanics for Materials and Structures*. Springer, 2018.

[70] M. Fleming, Y. A. Chu, B. Moran, T. Belytschko. Enriched element-free Galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering*, 40:1483–1504, 1997.

[71] M. Ghajari, L. Iannucci, P. Curtis. A peridynamic material model for the analysis of dynamic crack propagation in orthotropic media. *Computer Methods in Applied Mechanics and Engineering*, 276:431–452, 2014.

[72] M. Gunzburger, R.B. Lehoucq. A nonlocal vector calculus with application to nonlocal boundary value problems. *Multiscale modeling and simulation*, 8 (5):1581–1598, 2010.

[73] M. Gunzburger, R.B. Lehoucq. A nonlocal vector calculus with applications to nonlocal boundary value problems. *Multiscale Modeling and Simulation*, 8 (5):1581–1598, 2010.

[74] M. Hillman, J.S. Chen. An Accelerated, Convergent and Stable Nodal Integration in Galerkin Meshfree Methods for Linear and Nonlinear Mechanics. *International Journal for Numerical Methods in Engineering*, 107:603–630, 2015.

[75] M. Hillman, J.S. Chen, S.-W. Chi. Stabilized and variationally consistent nodal integration for meshfree modeling of impact problems. *Computational Particle Mechanics*, 1:245–256, 2014.

[76] R.-J Li M. J. Gander, Y.-L. Jiang. Parareal schwarz waveform relaxation methods. *Domain Decomposition Methods in Science and Engineering*, 60, Lecture Notes in Computational Science and Engineering:45–56, 2013.

[77] D.J. Littlewood M. Pasetto. Semi-Lagrangian RKPM and its Implementation in the Sierra/Solidmechanics Analysis Code. In *Center for Computing Research Summer Proceedings 2016, J.B. Carleton and M.L. Parks, eds., Technical Report SAND2017-1294R, Sandia National Laboratories*, pages 241–253, 2016.

[78] M. Pasetto, J.S. Chen, P. Seleson. Reproducing kernel enhanced peridynamics. Presentation part of the Computational and Applied Mathematics (CAM) seminar series at the Oak Ridge National Laboratory. Oak Ridge, TN, 2018.

[79] M. Pasetto, J.S. Chen, Y. Leng, J.T. Foster, P. Seleson. Reproducing kernel enhanced peridynamics. Presentation at the 13th World Congress on Computational Mechanics (WCCM2018). New York, NY, 2018.

[80] M. Pasetto, J.S. Chen, Y. Leng, J.T. Foster, P. Seleson. Reproducing kernel enhanced peridynamics. Presentation at the USACM Meshfree and Particle Methods: Applications and Theory conference. Santa Fe, NM, 2018.

[81] M. Pasetto*, Y. Leng*, J.S. Chen, J.T. Foster, P. Seleson, *contributed equally. A reproducing kernel enhanced approach for peridynamic solutions. *Computer Methods in Applied Mechanics and Engineering*, 340:1044–1078, 2018.

[82] J. Allen M. W. Reichelt, J.K. White. Optimal convolution sor acceleration of waveform relaxation with application to parallel simulation of semiconductor devices. *SIAM Journal on Scientific and Statistical Computing*, 16 (5):1137–1158, 1995.

[83] M.A. Bessa, J.T. Foster, T. Belytschko, W.K. Liu. A meshfree unification: reproducing kernel peridyamics. *Computational Mechanics*, 53:1251–1264, 2014.

[84] M.J. Gander. Overlapping Schwarz for Parabolic Problems. *Proceedings of the Ninth International Conference on Domain Decomposition Methods*, 1998.

[85] M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S.A. Silling. Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications*, 179:777–783, 2008.

[86] M.L. Wilkins, M.W Guinan. Impact of cylinders on a rigid boundar. *Journal of Applied Physics*, 44:1200–1206, 1973.

[87] N. Zhu, D. De Meo, E. Oterkus. Modeling of granular fracture in polycrystalline materials using ordinary state-based peridynamics. *Materials*, 9(12), 2016.

[88] N.Trask, H. You, Y. Yu, M.L. Parks. An asymptotically compatible meshfree quadrature rule for nonlocal problems with applications to peridynamics. *Computer Methods in Applied Mechanics and Engineering*, 343:151–165, 2019.

[89] O.-L. Kwok, P.-C. Guan, W.-P. Cheng, C.-T. Sun. Semi-Lagrangian Reproducing Kernel Particle Method for Slope Stability Analysis and Post-Failure Simulation. *KSCE Journal of Civil Engineering*, 19:107–115, 2015.

[90] O. Weckner, R. Abeyaratne. The effect of long-range forces on the dynamics of a bar. *Journal of the Mechanics and Physics of solids*, 53:705–728, 2005.

[91] P. Chartier, B. Philippe. A parallel shooting technique for solving dissipative odes. *Computing*, 51:209–236, 1993.

[92] P. Seleson. Improved one-point quadrature algorithms for two-dimensional peridynamic models based on analytical calculations. *Computer methods in applied mechanics and engineering*, 282:184–217, 2014.

[93] P. Seleson, D.J. Littlewood. Convergence studies in meshfree peridynamic simulations. *Computers and Mechanics with Applications*, 71:2432–2448, 2016.

[94] P. Seleson, D.J. Littlewood. Numerical tools for improved convergence of meshfree peridynamic discretizations. In *Handbook of Nonlocal Continuum Mechanics for Materials and Structures*. G. Voyiadjis ed., To Appear.

[95] P. Seleson, M. Parks. On the Role of the Influence Function in the Peridynamic Theory. *International Journal for Multiscale Computational Engineering*, 9(6):689–706, 2011.

[96] P. Seleson, Q. Du, M.L. Parks. On the consistency between nearest-neighbor peridynamic discretizations and discretized classical elasticity models. *Computational Methods in Applied Mechanics and Engineering*, 311:698–722, 2016.

[97] P.C. Guan, J.S. Chen, Y. Wu, H. Teng, J. Gaidos, K. Hofstetter, M. Alsaleh. Semi-Lagrangian reproducing kernel formulation and application to modeling earth moving operations. *Mechanics of Materials*, 41:670–683, 2009.

[98] P.C. Guan, S.W. Chi, J.S. Chen, T.R. Slawson, M.J. Roth. Semi-Lagrangian Reproducing Kernel Particle Method for Fragment-Impact Problems. *International Journal of Impact Engineering*, 38:1033–1047, 2011.

[99] Prashant K. Jha, Robert Lipton. Finite differences and finite elements in nonlocal fracture modeling: A priori convergence rates. In G. Voyiadjis, editor, *Handbook of Nonlocal Continuum Mechanics for Materials and Structures*. Springer, 2018.

[100] Q. Du, L. Ju, L. Tian, K. Zhou. A posteriori error analysis of finite element method for linear nonlocal diffusion and peridynamic models. *Mathematics of Computation*, 82(284):1889–1922, 2013.

[101] Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou. Analysis and approximation of nonlocal diffusion problems with volume constraints. *SIAM Review*, 54 (4):667–696, 2012.

[102] Q. Du, M. Gunzburger, R.B. Lehoucq, K.Zhou. A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. *Mathematical Models and Methods in Applied Science*, 23(3):493–540, 2013.

[103] Qiang Du. Nonlocal calculus of variations and well-posedness of peridynamics. In F. Bobaru, J.T. Foster, P.H. Geubelle and S.A. Silling, editor, *Handbook of Peridynamic Modeling*. CRC Press, 2017.

[104] R. Salehi, M. Dehghan. A generalized moving least square reproducing kernel method. *Journal of Computational and Applied Mathematics*, 249:120–132, 2013.

[105] K.J. Kim R.Cools. A survey of known and new cubature formulas for the unit disk. Technical report, Katholieke Universiteit Leuven, 2000.

[106] R.W. Macek, S.A. Silling. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design*, 43(15):1169–1178, 2007.

[107] S. Güttel. A parallel overlapping time-domain decomposition method for odes. *Domain Decomposition Methods in Science and Engineering XX*, pages 459–466, 2013.

[108] S. Li, W.K. Liu. *Meshfree Particle Methods*. Springer Verlag, 2004.

[109] H. Zhang S. Ta'asan. On the multigrid waveform relaxation method. *SIAM Journal on Scientific Computing*, 16 (5):1092–1104, 1995.

[110] S. Vandewalle, G. Horton. Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing*, 54 (317-330), 1996.

[111] S.A. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48:175–209, 2000.

[112] S.A. Silling, E. Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures*, 83:1526–1535, 2005.

[113] S.A. Silling, M. Epton, O. Weckner, J.Xu, E. Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88:151–184, 2007.

[114] S.A. Silling, O. Weckner, E. Askari, F. Bobaru. Crack nucleation in a peridynamic solid. *International Journal of Fracture*, 162:219–227, 2010.

[115] S.A. Silling, R.B. Lehoucq. Convergence of peridynamics to classical elasticity theory. *Journal of Elasticity*, 93:13–37, 2008.

[116] S.A. Silling, R.B: Lehoucq. Peridynamic theory of solid mechanics. *Advances in Applied Mechanics*, 44:73–168, 2010.

[117] S.F. Henke, S. Shanbhag. Mesh sensitivity in peridynamic simulations. *Computer Physics Communications*, 185:181–193, 2014.

[118] A.H. Stroud. Approximate calculation of multiple integrals. *American Mathematical Society*, 1971.

[119] S.W. Attaway, M.W. Heinstein, J.W. Swegle. Coupling of smooth particle hydrodynamics with the finite element method. *Nuclear Engineering and Design*, 150:199–205, 1994.

[120] S.W. Chi, C.H. Lee, J.S. Chen, P.C. Guan. A level set enhanced natural kernel contact algorithm for impact and penetration modeling. *International Journal for Numerical Methods in Engineering*, 102:839–866, 2015.

[121] T. Belytschko, D. Organ, Y. Krongauz. A coupled finite element-element free Galerkin method. *Computational Mechanics*, 17:186–195, 1995.

[122] T. Belytschko, M. Tabbara. Dynamic fracture using element?free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 39(3):923–938, 1995.

[123] T. Belytschko, W.K. Liu, B. Moran, K.I. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2nd edition, 2014.

[124] T. Belytschko, Y.Y. Lu, L. Gu. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics*, 51(2):295–315, 1995.

[125] T. Mengesha, Q. Du. Analysis of a scalar nonlocal peridynamic model with a sign changing kernel. *Discrete and continuous dynamical systems series B*, 18(5):1415–1437, 2013.

[126] T. Mengesha, Q. Du. The bond-based peridynamic system with dirichlet-type volume constraint. *Proceedings of the Royal Society of Edinburgh*, 144A:161–186, 2014.

[127] T. Rabczuk, S. P. Xiao, M. Sauer. Coupling of mesh-free methods with finite elements: basic concepts and test results. *Communications in Numerical Methods in Engineering*, 22:1031–1065, 2006.

[128] T. Siriaksorn, S.W. Chi, C. Foster, A. Mahdavi. u-p semi-Lagrangian reproducing formulation for landslide modeling. *International Journal for Numerical and Analytical Methods in Geomechanics*, 42:231–255, 2018.

[129] O. Nevanlinna U. Miekkala. Convergence of dynamic iteration methods for initial value problems. *SIAM Journal on Scientific and Statistical Computing*, 8-4:459–482, 1987.

[130] O. Nevanlinna U. Miekkala. Sets of convergence and stability regions. *BIT*, 27 (4):554–584, 1987.

[131] S. Vandewalle. *The Parallel Solution of Parabolic Partial Differential Equations by Multigrid Waveform Relaxation Methods*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, 1992.

[132] W. Gerstle, N. Sau, S.A. Silling. Peridynamic modeling of plain and reinforced concrete structures. In *18th International conference on structural mechanics in reactor technology (SMiRT 18). Beijing, China, no. SMiRT18-B01-2*. 2005.

[133] W. Gerstle, N. Sau, S.A. Silling. Peridynamic modeling of concrete structures. *Nuclear Engineering and Design*, 237:1250–1258, 2007.

[134] W. Hackbusch. Parabolic multi-grid methods. *Computing Methods in Applied Sciences and Engineering, VI*, pages 189–197, 1984.

[135] W. Hu, Y.D. Ha, F. Bobaru. Peridynamic model for dynamic fracture in unidirectional fiber-reinforced composites. *Computer Methods in Applied Mechanics and Engineering*, 217-220:247–261, 2012.

[136] W.K. Liu, R.A. Uras, Y. Chen. Enrichment of the finite element method with the reproducing kernel particle method. *Journal of Applied Mechanics*, 64:861–870, 1997.

[137] W.K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko. Reproducing Kernel Particle Methods for Structural Dynamics. *International Journal for Numerical Methods in Engineering*, 38:1655–1679, 1995.

[138] W.K. Liu, S. Jun, Y.F. Zhang. Reproducing Kernel Particle Methods. *International Journal for Numerical Methods in Fluids*, 20:1081–1106, 1995.

[139] W.L. Miranker, W. Liniger. Parallel methods for the numerical integration of ordinary differential equations. *Mathematics of Computation*, 21(99):303–320, 1967.

[140] X. Chen, M. Gunzburger. Continuous and discontinuous finite element methods for a peridynamics model of mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200:1237–1250, 2011.

[141] X. Tian, Q. Du. Analysis and comparison of different approximations to nonlocal diffusion and linear peridynamic equations. *Journal on Numerical Analysis*, 51(6):3458–3482, 2013.

[142] X. Tian, Q. Du. Nonconforming discontinuous Galerkin methods for nonlocal variational problems. *SIAM Journal on Numerical Analysis*, 53(2):762–781, 2015.

[143] Y. Hu, E. Madenci, N. Phan. Peridynamics Analysis of Impact Damage in Composite Laminates. *Journal of Aerospace Engineering*, 21(3):187–194, 2008.

[144] Y. Hu, E. Madenci, N. Phan. Peridynamics for predicting damage and its growth in composites. *Fatigue & Fracture of Engineering Materials & Structures*, 40:1214–1226, 2017.

[145] O. Wing Y.-L. Jiang, R.M.M Chen. Waveform relaxation of nonlinear second-order differential equations. *Circuits and Systems I: Fundamental Theory and Applications*, 48 (11):1344–1347, 2001.

[146] G. Turinici Y. Maday. The parareal in time iterative solver: a further direction to parallel implementation. *Domain Decomposition Methods in Science and Engineering*, 40, Lecture Notes in Computational Science and Engineering:441–448, 2005.

[147] Y.D. Ha, F. Bobaru. Studies of dynamic crack propagation and crack branching with peridynamics. *International Journal of Fracture*, 162:229–244, 2010.

[148] Y.D. Ha, F. Bobaru. Characteristics of dynamic brittle fracture captured with peridynamics. *Engineering Fracture Mechanics*, 78:1156–1168, 2011.

[149] Y.L. Hu, N.V. De Carvalho, E. Madenci. Peridynamic modeling of delamination growth in composite laminates. *Composite Structures*, 132:610–620, 2015.