

Mechanical Behavior of Materials at Multiscale  
Peridynamic Theory and Learning-based Approaches

by

Sayna Ebrahimi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering — Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor David Steigmann, Co-chair

Professor Trevor Darrell, Co-chair

Professor Lisa Pruitt

Professor Robert Ritchie

Spring 2020

**Mechanical Behavior of Materials at Multiscale  
Peridynamic Theory and Learning-based Approaches**

Copyright 2020  
by  
Sayna Ebrahimi

## Abstract

### Mechanical Behavior of Materials at Multiscale Peridynamic Theory and Learning-based Approaches

by

Sayna Ebrahimi

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor David Steigmann, Co-chair

Professor Trevor Darrell, Co-chair

Classical continuum mechanics has been widely used in the failure analysis of materials for decades. However, spatial partial derivatives in governing equations of the conventional theory are not valid along the discontinuities. Alternatively, peridynamic (PD) theory, as a nonlocal continuum theory, eliminates this shortcoming by using integro-differential equations which do not contain spatial derivatives. This feature makes PD theory very attractive for problems including discontinuities such as cracks.

In this study we have extended PD formulation to multiscale problems involving friction, wear, and delamination of thin films. We have formulated a nonlocal ordinary state-based peridynamic formulation for plastic deformation based on the idea of mechanical sublayers which is successfully applied in modeling ductile fracture. In addition, we demonstrated how PD can be used as an efficient and accurate analysis tool in designing real-world applications such as body armor systems using bio-inspired structures with the goal of minimizing the effect of the ballistic impact and bullet penetration depth while being lightweight and comfortable to wear. All our obtained results are validated against experimental observations and an excellent agreement has been achieved. Similar to other mesh-free methods, PD is massively parallelizable. We built parallel PD algorithms leveraging shared and distributed memory systems on CPU as well as CUDA architecture on GPU. We provide extensive experiments showing scalability and bottlenecks associated with each parallelization technique.

In the second part of this dissertation, we introduce a new class of learnable forward and inference models, using graph neural networks (GNN) which develops relational behavior between material points. We demonstrate these models are surprisingly accurate to generalize remarkably well to challenging unseen loading conditions. Our framework offers new opportunities for harnessing and exploiting non-local continuum theory and powerful statistical learning frameworks to take a key step toward building accurate, robust, and efficient patterns of reasoning about materials behavior.

To my parents, Maryam and Hossein.

Thank you.

# Contents

|                                                                           |           |
|---------------------------------------------------------------------------|-----------|
| <b>Contents</b>                                                           | <b>ii</b> |
| <b>List of Figures</b>                                                    | <b>v</b>  |
| <b>List of Tables</b>                                                     | <b>ix</b> |
| <b>1 Introduction</b>                                                     | <b>1</b>  |
| <b>I Peridynamic Analysis of Materials Behavior</b>                       | <b>7</b>  |
| <b>2 Introduction to Peridynamic Theory</b>                               | <b>8</b>  |
| 2.1 Preliminaries . . . . .                                               | 8         |
| 2.2 State-based Peridynamic Constitutive Equation . . . . .               | 10        |
| 2.3 Numerical solution scheme . . . . .                                   | 14        |
| 2.4 Peridynamic pseudocode . . . . .                                      | 15        |
| 2.5 Original Formulation . . . . .                                        | 16        |
| 2.5.1 Body discretization and computational details . . . . .             | 17        |
| <b>3 Studying Wear and Friction in Carbon Thin Films with Peridynamic</b> | <b>19</b> |
| 3.1 Amorphous carbon thin films . . . . .                                 | 19        |
| 3.2 Peridynamics friction and wear models . . . . .                       | 20        |
| 3.2.1 Friction model . . . . .                                            | 21        |
| 3.2.2 Wear model . . . . .                                                | 22        |
| 3.3 Results and discussion . . . . .                                      | 24        |
| 3.3.1 Coefficient of friction . . . . .                                   | 25        |
| 3.3.2 Wear depth . . . . .                                                | 27        |
| <b>4 Peridynamic Modeling of Delamination in Thin Compressed Films</b>    | <b>31</b> |
| 4.1 Introduction to Delamination . . . . .                                | 31        |
| 4.1.1 Deformation mechanisms in compressed thin films . . . . .           | 32        |
| 4.1.2 Why Peridynamic? . . . . .                                          | 32        |
| 4.2 Modeling deformation mechanisms in thin films . . . . .               | 33        |

|          |                                                               |           |
|----------|---------------------------------------------------------------|-----------|
| 4.2.1    | Wrinkling without delamination . . . . .                      | 33        |
| 4.2.1.1  | Analytical solution . . . . .                                 | 34        |
| 4.2.1.2  | Peridynamic delamination model . . . . .                      | 35        |
| 4.2.2    | Effect of mechanical properties mismatch . . . . .            | 37        |
| 4.2.3    | Wrinkling with delamination . . . . .                         | 38        |
| 4.3      | Surface pattern prediction . . . . .                          | 39        |
| 4.4      | Discussion . . . . .                                          | 42        |
| <b>5</b> | <b>Peridynamic for Plastic Deformation</b>                    | <b>44</b> |
| 5.1      | Plastic deformation . . . . .                                 | 45        |
| 5.1.1    | Peridynamics plasticity using sublayer method . . . . .       | 45        |
| 5.2      | Benchmark problems . . . . .                                  | 48        |
| 5.2.1    | Center cracked plate in tension . . . . .                     | 48        |
| 5.2.2    | Compact tension . . . . .                                     | 51        |
| 5.2.3    | Summary . . . . .                                             | 53        |
| <b>6</b> | <b>Peridynamic for Ballistic Impact and Body Armor Design</b> | <b>55</b> |
| 6.1      | Introduction . . . . .                                        | 55        |
| 6.1.1    | Body armor . . . . .                                          | 55        |
| 6.2      | Body armor design . . . . .                                   | 57        |
| 6.2.1    | Rigid impactor . . . . .                                      | 57        |
| 6.2.2    | Non-rigid impactor . . . . .                                  | 60        |
| 6.3      | Performance analysis . . . . .                                | 61        |
| <b>7</b> | <b>Parallel Algorithms for Peridynamic</b>                    | <b>64</b> |
| 7.1      | Peridynamic is massively parallelizable . . . . .             | 64        |
| 7.2      | Algorithms and Implementations . . . . .                      | 65        |
| 7.2.1    | Serial algorithm . . . . .                                    | 66        |
| 7.2.2    | Serial implementation details . . . . .                       | 67        |
| 7.2.3    | Parallel algorithms . . . . .                                 | 68        |
| 7.2.4    | Shared-memory algorithm: OpenMP . . . . .                     | 68        |
| 7.2.5    | OpenMP implementation details . . . . .                       | 69        |
| 7.2.6    | Distributed memory algorithm: MPI . . . . .                   | 70        |
| 7.2.7    | MPI implementation details . . . . .                          | 71        |
| 7.2.8    | GPU: CUDA . . . . .                                           | 72        |
| 7.2.9    | CUDA implementation details . . . . .                         | 73        |
| 7.3      | Results and discussion . . . . .                              | 74        |
| 7.3.1    | Speed up . . . . .                                            | 74        |
| 7.3.2    | Scaling efficiency . . . . .                                  | 78        |

|           |                                                                     |           |
|-----------|---------------------------------------------------------------------|-----------|
| <b>II</b> | <b><i>Learning</i> Mechanical Behavior of Materials</b>             | <b>83</b> |
| <b>8</b>  | <b>Materials Modeling via Graph Neural Networks</b>                 | <b>84</b> |
| 8.1       | Introduction to Neural Networks . . . . .                           | 84        |
| 8.1.1     | Feed-forward Neural Networks . . . . .                              | 85        |
| 8.2       | Graph Representation of a Material . . . . .                        | 86        |
| 8.3       | Experimental Setup . . . . .                                        | 89        |
| 8.4       | Results . . . . .                                                   | 90        |
| <b>9</b>  | <b>Conclusion and Future Work</b>                                   | <b>94</b> |
| 9.1       | Discussion of Contributions . . . . .                               | 94        |
| 9.1.1     | Part I: Peridynamic Analysis of Materials Behavior . . . . .        | 94        |
| 9.1.2     | Part II: <i>Learning</i> Mechanical Behavior of Materials . . . . . | 96        |
| 9.2       | Future Perspectives . . . . .                                       | 96        |
|           | <b>Bibliography</b>                                                 | <b>98</b> |

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Family members of $x_i$ and $x_j$ centered at their horizons with radii of $\delta$ share a bond in two-dimensional Cartesian space . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 9  |
| 2.2 | PD approaches: bond-based theory, ordinary state-based theory, non-ordinary state based theory . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 10 |
| 3.1 | Schematic of peridynamics friction model of a spherical diamond (rigid) tip sliding at constant velocity $V$ against a thin $a$ -C film, which is firmly adhered to a thick Si substrate. The tip slides from left ( $x/S = 0$ ) to right ( $x/S = 1$ ) by a total distance $S$ . The shaded layer at the bottom of the substrate is modeled as rigid. The coefficient of friction is obtained as the ratio of the computed tangential (friction) force $F$ , which opposes tip sliding, and the applied normal load $P$ . The film thickness and the tip radius are not drawn to scale. . . . . | 22 |
| 3.2 | Schematic of peridynamics wear model of a sharp conospherical diamond (rigid) tip under a normal load $P$ sliding at constant velocity $V$ and plowing through a thin $a$ -C film, which is firmly attached to a thick Si substrate. The tip slides against the film surface from left ( $x/S = 0$ ) to right ( $x/S = 1$ ) by a total distance $S$ . The shaded layer at the bottom of the substrate is modeled as rigid. The film thickness and the tip radius are not drawn to scale. . . . .                                                                                                 | 24 |
| 3.3 | Peridynamic and experimental results of the coefficient of friction of an $a$ -C film versus sliding distance for $P = 400 \mu\text{N}$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                | 27 |
| 3.4 | Peridynamic results of the coefficient of friction of various $a$ -C films versus normal load. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 28 |
| 3.5 | (a), (b) Contour maps of $y$ - and $x$ -displacements of particles in film #4 for $x/S = 1$ and (c), (d) corresponding high-magnification contour maps showing the displacement of particles in the near-surface region of the film adjacent to the contact interface with the sliding rigid tip. . . . .                                                                                                                                                                                                                                                                                        | 29 |
| 3.6 | (a) Damage contour map of film #9 after tip unloading (damage index $D = 1$ corresponds to particles with all their bonds broken), (b) corresponding high-magnification damage contour map of damaged particle layers in the near-surface region of the film, (c) $y$ -displacement contour map of damaged film #9 after tip unloading, and (d) corresponding high-magnification contour map showing the $y$ -displacement of particles adjacent to the contact interface with the plowing rigid tip. . . . .                                                                                    | 30 |



|      |                                                                                                                                                                                                                                                                |    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.1  | Deformation mechanisms in compressed thin films. From left to right: wrinkling with no delamination, buckling-driven interface delamination, concomitant wrinkling and buckle delamination . . . . .                                                           | 32 |
| 4.2  | Controlling parameters for surface patterns on the thin compressed film on a compliant substrate . . . . .                                                                                                                                                     | 34 |
| 4.3  | Schematic of the thin film and the compliant substrate model . . . . .                                                                                                                                                                                         | 35 |
| 4.4  | Comparison between analytical solution and PD for the critical strain required to onset wrinkling without delamination for various Poisson's ratios of the substrate.                                                                                          | 36 |
| 4.5  | Effect of elastic moduli mismatch on onset of wrinkling with no delamination for different film and substrate thicknesses ratios . . . . .                                                                                                                     | 37 |
| 4.6  | Effect of critical bond stretch mismatch on onset of wrinkling with no delamination for different film and substrate thicknesses ratios . . . . .                                                                                                              | 37 |
| 4.7  | Effect of layers' thickness mismatch on onset of wrinkling with no delamination for different elastic moduli mismatches . . . . .                                                                                                                              | 38 |
| 4.8  | Comparison between analytical solution and PD for the critical strain required to onset wrinkling with delamination for various Poisson's ratios of the substrate.                                                                                             | 39 |
| 4.9  | Effect of elastic moduli mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios . . . . .                                                                                                                        | 40 |
| 4.11 | Energy density function for film/substrate undergoing compressive strain until wrinkling pattern generation . . . . .                                                                                                                                          | 40 |
| 4.10 | Effect of critical bond stretch mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios . . . . .                                                                                                                 | 41 |
| 4.12 | Energy density function for film/substrate undergoing compressive strain exhibiting concomitant wrinkling and buckling behavior . . . . .                                                                                                                      | 41 |
| 4.13 | Effect of Poisson's ratio mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios . . . . .                                                                                                                       | 42 |
| 4.14 | . . . . .                                                                                                                                                                                                                                                      | 43 |
| 5.1  | (a) Uniaxial stress-strain curve ; (b) Constructing sub-layer method with three layers . . . . .                                                                                                                                                               | 45 |
| 5.2  | (a) Sub-layers of a single peridynamic bond undergoing force $\hat{\mathbf{t}}$ ; (b) Constructing $n$ layers and recording their critical stretch values at which they yield. $e_1$ is the Young's modulus ( $E$ ) of the material. . . . .                   | 46 |
| 5.3  | (a) A finite center-cracked plate with crack size of $2a$ , and tensile displacement boundary condition made from ferrite steel; (b) Stress-strain curve for ferrite steel given in Eq. (5.7) and constructing sublayers with selected stress-strain pairs . . | 47 |
| 5.4  | Comparison of PD J-integral vs. experimental results by (Gullerud <i>et al.</i> , 2000)                                                                                                                                                                        | 49 |
| 5.5  | Bond force vs. stretch for a single bond undergoing loading and unloading . . .                                                                                                                                                                                | 50 |
| 5.6  | Subsequent displacement contours in the $y$ -direction for (a) $\bar{v} = 0.704$ mm, (b) $\bar{v} = 0.985$ mm, (C) $\bar{v} = 6.471$ mm . . . . .                                                                                                              | 50 |
| 5.7  | Subsequent velocity contours in the $y$ -direction for (a) $\bar{v} = 1.270$ mm, (b) $\bar{v} = 2.813$ mm, (C) $\bar{v} = 3.871$ mm . . . . .                                                                                                                  | 51 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.8  | Subsequent yield index contours for (a) $\bar{v} = 1.270$ mm, (b) $\bar{v} = 2.813$ mm, (c) $\bar{v} = 4.572$ mm . . . . .                                                                                                                                                                                                                                                                                                                                                                            | 52 |
| 5.9  | (a) CT specimen geometry from (Areias <i>et al.</i> , 2014) (b) stress-strain curve for the material used in the CT specimen from (Areias <i>et al.</i> , 2014) . . . . .                                                                                                                                                                                                                                                                                                                             | 53 |
| 5.10 | J-integral calculation comparison for the CT specimen . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                       | 54 |
| 5.11 | Yield index contour for the CT specimen for (a) $\bar{v} = 1.99$ mm and (b) $\bar{v} = 2.82$ mm . . . . .                                                                                                                                                                                                                                                                                                                                                                                             | 54 |
| 6.1  | Schematic of the dual-material ballistic protection system . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                  | 57 |
| 6.2  | From left to right: 1)Initial configuration of WC-Co projectile (blue) and conventional body armor consisting of a SiC outer layer (green) and a homogeneous polyethylene backing (yellow), 2) deformed conventional body armor with buried projectile, 3) initial configuration of WC-Co projectile (blue) and new body armor design with a SiC outer layer (black) and epoxy-layered backing with strong interfaces (multi-colored), and 4) deformed new body armor with buried projectile. . . . . | 59 |
| 6.3  | Bullet's velocity profile during the penetration . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                            | 59 |
| 6.4  | Velocity profile of different projectiles from Table 6.3 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                    | 63 |
| 6.5  | Displacement contour in vertical direction for design pattern #11. (a) bullet exits the front layer and penetrates thoroughly into it after 50 $\mu$ s(b) The second layer successfully envelops the bullet by absorbing its kinetic energy after 75 $\mu$ s (c) Unloading occurs and the bullet is pushed backward by small amount after 375 $\mu$ s . . . . .                                                                                                                                       | 63 |
| 7.1  | A distributed-memory system . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 68 |
| 7.2  | A shared-memory system . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 69 |
| 7.3  | Center-cracked plate under velocity boundary condition for testing the parallel algorithms . . . . .                                                                                                                                                                                                                                                                                                                                                                                                  | 75 |
| 7.4  | log-log plot of simulation time for different number of particles for the naive serial code and the modified one with binning method. . . . .                                                                                                                                                                                                                                                                                                                                                         | 76 |
| 7.8  | log-log plot of simulation time versus different number of particles for naive and optimized CUDA GPU . . . . .                                                                                                                                                                                                                                                                                                                                                                                       | 76 |
| 7.5  | log-log plot of simulation time versus different number of particles for a fixed number of threads of 4 (top) and 24 (bottom) . . . . .                                                                                                                                                                                                                                                                                                                                                               | 77 |
| 7.6  | log-log plot of simulation time versus different number of threads for a fixed number of particles = 50000 . . . . .                                                                                                                                                                                                                                                                                                                                                                                  | 78 |
| 7.9  | Slopes estimates for Algorithm 3 implementation . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                             | 78 |
| 7.7  | Comparison of Mflops/sec versus different number of threads for OpenMP and MPI . . . . .                                                                                                                                                                                                                                                                                                                                                                                                              | 79 |
| 7.10 | OpenMP scaling: from top image to below: strong scaling of speed up, strong scaling of efficiency, weak scaling of efficiency . . . . .                                                                                                                                                                                                                                                                                                                                                               | 81 |
| 7.11 | MPI scaling: from top image to below: strong scaling of speed up, strong scaling of efficiency, weak scaling of efficiency . . . . .                                                                                                                                                                                                                                                                                                                                                                  | 82 |
| 8.1  | A multi-layer perceptron (MLP) with a single hidden layer . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                   | 86 |

|     |                                                                                                                                                                                                                                                                                   |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 8.2 | A graph sample with $\mathbf{v}_i$ as a node, $\mathbf{e}_k$ as an edge, and $\mathbf{g}$ as the global attributes such as dynamic force which can be encoded as a vector or even another graph.                                                                                  | 87 |
| 8.3 | (a) A <i>graph2graph</i> network takes a graph as input and returns a graph with the same structure but different edge, node, and global features as output. (c) A feed-forward GN-based forward model for learning one-step predictions. (d) A recurrent GN-based forward model. | 88 |
| 8.4 | Damage index per particle for a plate with no pre-existing defect. (a) shows the initial plate at time $t = 0$ . (b-e) damage at timesteps $t = 5\Delta t$ , $t = 15\Delta t$ , $t = 20\Delta t$ , $t = 25\Delta t$ , respectively                                                | 91 |
| 8.5 | Damage index per particle for a plate with randomly placed pre-existing holes of different size. (a) shows the initial plate at time $t = 0$ . (b-e) damage at timesteps $t = 5\Delta t$ , $t = 15\Delta t$ , $t = 20\Delta t$ , $t = 25\Delta t$ , respectively                  | 92 |
| 8.6 | Yield index per particle for a plate with randomly placed pre-existing holes of different size. (a) shows the initial plate at time $t = 0$ . (b-e) yield effect at timesteps $t = 5\Delta t$ , $t = 15\Delta t$ , $t = 20\Delta t$ , $t = 25\Delta t$ , respectively             | 92 |
| 8.7 | Yield index per particle for pre-cracked plate. (a) shows the initial plate at time $t = 0$ . (b-e) yield effect at timesteps $t = 5\Delta t$ , $t = 15\Delta t$ , $t = 20\Delta t$ , $t = 25\Delta t$ , respectively                                                             | 93 |

# List of Tables

|     |                                                                                                                                                                                                                   |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Peridynamic material constants . . . . .                                                                                                                                                                          | 13 |
| 3.1 | Mechanical properties and critical stretch of Si substrate. <sup>(a)</sup> Ref. (Lu & Komvopoulos, 2001) and <sup>(b)</sup> Ref. (Agwai <i>et al.</i> , 2011) . . . . .                                           | 21 |
| 3.2 | Thickness, roughness, elastic properties, and density of <i>a</i> -C films and numbers of film and substrate particles used in the peridynamics friction analysis (Lu & Komvopoulos, 2001). . . . .               | 23 |
| 3.3 | Thickness, roughness, elastic properties, density, and critical stretch of <i>a</i> -C films and numbers of film and substrate particles used in the peridynamics wear analysis (Lu & Komvopoulos, 2001). . . . . | 23 |
| 3.4 | Coefficient of friction results from <i>m</i> -convergence tests. . . . .                                                                                                                                         | 25 |
| 3.5 | Peridynamics and experimental results of the coefficient of friction of <i>a</i> -C films versus film thickness, roughness, and normal load. . . . .                                                              | 26 |
| 3.6 | Comparison of peridynamics and experimental results of the wear depth of <i>a</i> -C films versus film thickness, roughness, and normal load . . . . .                                                            | 28 |
| 4.1 | Material properties of the film and substrate in analysis of deformation mechanisms                                                                                                                               | 36 |
| 5.1 | Material properties of ferritic steel in center-cracked plate used by (Gullerud <i>et al.</i> , 2000) . . . . .                                                                                                   | 48 |
| 5.2 | Weighting factors and critical bond stretch values for yield prediction in compact tension specimen . . . . .                                                                                                     | 48 |
| 5.3 | Material properties of compact tension specimen reported in (Areias <i>et al.</i> , 2014)                                                                                                                         | 51 |
| 5.4 | Weighting factors and critical bond stretch values for yield prediction in center-cracked plate . . . . .                                                                                                         | 52 |
| 6.1 | Mechanical properties of materials used in the simulation . . . . .                                                                                                                                               | 58 |
| 6.2 | Comparison between the mass and velocity reduction of the projectile in two models                                                                                                                                | 60 |
| 6.3 | Possible combinations of three different projectiles with two different ceramic-based front layers and polymeric backing layers. . . . .                                                                          | 62 |
| 6.4 | Material properties used in simulating front layer, backing layer and the projectile                                                                                                                              | 62 |
| 7.1 | Material properties and PD parameters of the center-cracked plate shown in Fig. 7.3 . . . . .                                                                                                                     | 74 |

|     |                                                |    |
|-----|------------------------------------------------|----|
| 7.2 | Scaling factors for OpenMP algorithm . . . . . | 79 |
| 7.3 | Scaling factors for MPI algorithm . . . . .    | 80 |

## Acknowledgments

The past few years have brought many people across my path, and all have contributed to my journey and accomplishment in different ways. It is important for me to recount and recognize how so many people played a role in the completion of this dissertation.

First and foremost, I want to express my sincere gratitude to both of my PhD co-advisors, Professor Trevor Darrell and Professor David Steigmann for taking me in as their advisee.

Trevor trusted my research ability since we met despite my different background and changed my life forever by accepting me to be part of his research group. He guided my research and challenged me to keep my theoretical mind sharp. He was always interested, invested, and actively involved in my research and was always willing to give comments on, be criticism of, and provide suggestion to improve any work I had done. I learned from him how to think broad, yet not losing focus from the ultimate long-term agenda. His consistent reminder for being confident and remaining strong has made me the researcher I am today. One of the biggest strengths Trevor has instilled in me is to not give up. He has been indeed one of the reasons I have kept fighting my cancer during all these years. I cannot thank him enough for being there for me not just as an advisor, but like a caring family member despite being responsible for leading such a large research group. I could always count on him to help me find a way out of any problem. I am thankful for his continued support and guidance as I embark on my academic journey ahead.

Over the years, David has always given me the freedom to explore my passion, even if they were only tangentially related to the main research problem at hand; he always had complete faith in my ability as a researcher and trusted that following my passion would only lead me to achieve more.

I would also like to thank Professor Lisa Pruitt who has been there as a teacher, mentor, and a committee member for the qualifying exam and dissertation. I was also her teaching assistant in ME 108 which was the most educating semester among the 6 times I co-thought this course. Her continued support always cheered me up during the most difficult times. I would also like to thank Professor Robert Ritchie for serving on my qualification exam committee and my dissertation committee. All I know about fracture mechanics is because of his excellent course. I should also thank Professor James Casey and Professor Ronald Gronskey for serving on my qualification exam committee. I was fortunate enough to be Professor Gronskey's student in the very last semester he taught MSE 205. I also like to thank Professor Kyriakos Komvopoulos who was my research co-advisor in the first year of my Ph.D. program. I am indebted to him for giving me the freedom to explore other research fields and transferring to Prof. Steigmann's group.

I had the pleasure of working with Scott McCormick, Michael Neuffer, Alex Jordan, and Pete Graham for 6 semesters teaching ME 108 and 107 labs. These gentlemen do a terrific job supporting, maintaining, and improving student labs. From day one they made me feel a part of their team and I am grateful for everything they thought me as a GSI and always being there for me to help with any problem in the lab.

I cannot thank enough Marcus Rohrbach who has been a tremendous help to me throughout my graduate career. Dr. Rohrbach as my supervisor during my internship at Facebook AI Research has been an amazing mentor and taught me great many things that shaped the foundations of my research. I am also thankful to Mohamed Elhoseiny, Franziska Meier, Roberto Calandra for their hospitality and advices during my internships at Facebook.

I am thankful to my labmates, Samaneh Azadi, Anna Rohrbach, Suzie Petryk, Seth Park, Devin Guillory, Erin Grant, Eric Tzang, Juddy Hoffman, Lisa Hendricks, Yang Gao, Parsa Mahmoudieh, Evan Shelhamer, Xin Wang, and Kate Rakelly. In addition to the people that directly informed, guided, and funded my research, many other people greatly helped me in completing this dissertation. First of all, I would like to thank my husband, Behrooz Shahsavari, for his constant understanding and support of my academic goals; he has been an enormous support for me and words cannot express my gratitude for him always being there for me. Last and foremost, I am thanking my parents, my sister and my brother who have given me unconditional love, support and encouragement throughout my life.

# Chapter 1

## Introduction

Studying behavior of materials under various loading conditions is a long-standing problem in mechanics. Researchers have used both experimental and numerical approaches to study mechanics of materials. Despite the great benefits of conducting real-world experiments on objects to explore the outcomes, it is not always feasible or practical to perform such experiments due to many reasons such as time, expense, or impossibility of conducting a perfectly controlled experiment to study one behavior at the time especially when we are dealing with multi-scale problems. As a result, there is always a great need for proposing computational approaches which represent the behavior of the materials based on well-established theoretical models.

Classical continuum theory, is the most well-known framework to study the physics of a continuous media which was first formulated by the great Cauchy in the early 19th century (Cauchy, 1822). Several numerical approaches such as Finite Element Method (FEM), e.g., (Rockey *et al.*, 1975; Zienkiewicz *et al.*, 1977; Banks-Sills, 1991), Boundary Element Methods (BEM), e.g., (Aliabadi & Rooke, 1991; Ebrahimi & Phan, 2013; Ebrahimi & Phan, 2015), Finite Difference Methods (FDM), e.g., (Mitchell & Griffiths, 1980), and particle based methods, e.g., (Silling, 2000) have been introduced to solve the governing equations which are derived by discretization of materials with *elements*, *nodes*, *lattices*, or *particles*.

Despite of the gained success in the process of designing engineering materials with complex geometries under mixed-mode loading conditions, they tend to break down when the strong assumptions of the field are no longer valid. First assumption that the classical theory is based upon is *continuity*. As the name suggests, the classical continuum theory assumes the material is continuous at all time, which is indeed not true as there exists no defect-free material in the world. In the presence of discontinuities, governing partial differential equations (PDE) in the classical continuum theory break down because they simply do not exist at singular points where the stress field is no longer differentiable.

The second strong assumption in the realm of the classical theory, is *locality*. At high level, locality means that a material point interacts merely with its immediate neighbors and therefore, long-range forces are neglected in this theory. Whereas the atomic theory has proven their existence (Madenci & Oterkus, 2014). In other words, all materials have



internal forces that are not just *contact* forces and their interactions occur across longer horizons (Bobaru *et al.*, 2016). This results in the classical theory becoming incapable of multi-scale failure analysis which occurs in all structures used in different applications.

Various modifications have been proposed over the past decades to mitigate the drawbacks of the classical approaches which are listed in the following: the idea of Cohesive Zone Model (CZM) (Dugdale, 1960) and Cohesive Zone Elements (CZE) (Hillerborg *et al.*, 1976) which made the calculations of crack initiation and crack propagation possible for both ductile and brittle materials. In fact they could overcome one of the shortcomings of the standard FEM which is incapability of predicting crack initiation as the crack must exist in the material for FEM to be able to analyze it and predict its direction which is not a realistic approach in failure analysis. However, the mesh-dependency of this method and unpredictable consequences of having a non-suitable mesh for a given problem, make them inefficient especially when location of the crack is not known a priori. Another worth-mentioning technique that has been developed to resolve difficulties associated with FEM, is eXtended Finite Element Method (XFEM) (Belytschko & Black, 1999) which allowed for crack propagation on an element surface instead of element boundaries which also surpassed CZE in eliminating the need for defining specific direction for crack to grow. Nonetheless, the XFEM solution is not accurate when multiple cracks interact together as the accuracy decreases in the elements adjacent to the element in which the crack tip lives. Moreover, XFEM still requires an artificial crack growth criterion to allow for it to occur.

To be at opposite extremes, another great framework came to exist to address the weaknesses of the classical continuum theory called Molecular Dynamics (MD). Although MD simulation happens to be the most realistic way of structural analysis, they are mainly used to study the basics of failure process rather than predicting and preventing it from occurrence and the main reason for that is it is yet not computationally possible to study a structure at atomic scale as a whole. Good news is that it is possible to get the best of both worlds by bridging the gap between the classical continuum theory and molecular dynamics by means of *non-local* continuum models where a material point is assumed to be interacting to other points within a finite horizon, making it possible for long-range forces to exist in this framework without restricting the material points to be as small as an atom size. The latter makes the non-local model to be less computationally demanding.

Various versions of non-local theories have been proposed to solve wide range of problems in mechanics. Eringen focused on this topic in most of his work such as (Eringen, 1972a; Eringen, 1972b; Eringen & Kim, 1974; Eringen *et al.*, 1977) in which they used their non-local approach for crack growth prediction by defining finite stresses at discontinuities. However similar to other non-local methods that live in the realm of the classical theory, they suffer from using spatial derivatives in their formulation which causes failure at the presence of discontinuities. (Kunin, 1982; Rogula, 1982) tried to use displacement fields and not their derivatives but it was only able to cover problems in one dimensional space. Later (Kunin, 1983) introduced a three dimensional framework by using a discrete lattice structure as an approximation of the continuous object.

After all, a new non-local theory, so-called Peridynamic, was introduced by (Silling,

2000) which could address all the aforementioned problems in other non-local theories. It uses the idea of using displacements instead of their derivatives similar to (Kunin, 1982) and reformulates the equations in the classical theory from PDEs to spatial integral equations which are valid at all points including discontinuities. The most important attribute of this theory is that *damage* is part of the solution, not a handcrafted rule assigned by the user to allow the discontinuity to occur and grow. This incredible feature allows this theory to be able to predict initiation and growth of damage of any kind at any scale. There is no need to have any knowledge about where the damage occurs a priori. Damage simply takes place when and where it means to happen.

Since the numerical solution of peridynamic equations of motion is computationally more expensive than the local solutions, such as FEM, it may be advantageous to combine PD theory and local solutions. In a recent study, (Seleson *et al.*, 2013) proposed a force-based blended model that coupled PD theory and classical elasticity by using nonlocal weights composed of integrals of blending functions. They also generalized this approach to couple peridynamics and higher order gradient models of any order. In another study, (Lubineau *et al.*, 2012) performed coupling of local and nonlocal solutions through a transition (morphing) that affects only constitutive parameters. The definition of the morphing functions in their approach relies on energy equivalence. In addition to these techniques, (Kilic & Madenci, 2010) and (Liu & Hong, 2012) coupled FEM and peridynamics. A more straightforward coupling procedure is given in (Macek & Silling, 2007), where the PD interactions are represented by truss elements. If only some part of the region is desired to be modeled by using peridynamics, then the other sections can be modeled by traditional finite elements. Another simple approach, demonstrated by (Oterkus & Madenci, 2012) and (Agwai *et al.*, 2012), was first to solve the problem by using finite element analysis and obtain the displacement field. Then, by using the available information, the displacements can be applied as a boundary condition to the peridynamic model of a critical region.

The peridynamic theory has been utilized successfully for damage prediction of many problems at different length scales from macro to nano. In order to take into account the effect of van der Waals interactions, (Silling & Bobaru, 2005) and (Bobaru *et al.*, 2005) included an additional term to the peridynamic response function to represent van der Waals forces. This new formulation was used to investigate the mechanical behavior, strength, and toughness properties of three-dimensional nanofiber networks under imposed stretch deformation. It was found that the inclusion of van der Waals forces significantly changes the overall deformation behavior of the nanofiber network structure. In a recent study, (Seleson *et al.*, 2009) demonstrated that peridynamics can play the role of an upscale version of molecular dynamics and pointed out the extent where the molecular dynamics solutions can be recovered by peridynamics.

Even though numerous journal articles and conference papers exist in the literature on the evolution and application of the peridynamic theory, it is still new to the scientific community. Because it is based on concepts not commonly used in the past.

Intuitively, more fine-grained discretizations result in more accurate solutions. Therefore, s Similar to all numerical approaches, Peridynamics also benefit from development of com-

putational power. Advances in computing power, not only supports the practicality of the existing approaches in continuum theory, but also have been the key factor in emerging data-driven and machine learning-based approaches (Samuel, 1959; Murphy, 2012), particularly deep learning methods (LeCun *et al.*, 2015). Although the core idea of these approaches was introduced a few decades ago (Samuel, 1959), researchers discontinued using them due to their scalability issues. However, with the striking progress of graphical processing units (GPU), deep learning in particular won overwhelming victory over any other technique in various fields such as computer vision (Krizhevsky *et al.*, 2012; He *et al.*, 2016), medical and healthcare applications (Lee *et al.*, 2017), autonomous driving (Ebrahimi *et al.*, 2017), generating high resolution images (Dong *et al.*, 2014; Azadi *et al.*, 2018), and is being widely used for collecting and analyzing enormous amount of data.

This work is concerned with driving deep into the peridynamic theory and deriving models and numerical analysis of different phenomena in mechanics. Peridynamic is relevantly a new research area and there is a lot of ongoing research in this framework. In particular we have tried to derive new models that enable us to use this framework for a variety of important applications in which the classical theory fails or performs poorly. Particularly we are interested in using peridynamics for modeling friction, wear, wrinkling, delamination, ballistic impact and deriving a plasticity theory in peridynamic framework. Applicability of PD on these phenomena significantly depends on the PD implementation efficiency. This has been also one of our biggest concern in this work. We designed various algorithms to run PD algorithm in parallel.

Here is a breakdown of this dissertation in terms of our contributions presented in each chapter:

- **Chapter 2:** This chapter will serve as an introduction to the peridynamic formulation including definitions and derivations. Currently there are two different formulations used to define concepts in PD theory. The first one with which the theory was originally introduced is (Silling, 2000) and the second one was developed by (Madenci & Oterkus, 2014). Both formulations are defined in this chapter.
- **Chapter 3:** This chapter covers modeling tribological phenomena (wear and friction) using Peridynamic theory. Our contribution in this chapter is using PD at nano-scale to compute coefficient of friction in sliding contact of a rigid tip over an ultra-thin film. We were also able to compute the penetration depth during such contact and the resulting material removal caused by it. The excellent agreement between the PD solution and the experimental setup we compared with, shows how PD can be promising in modeling such contact mechanisms where classical theory fails at.
- **Chapter 4:** We introduce a new PD model to predict three modes of delamination in compressed elastic thin films on compliant substrates: wrinkling, buckling and concomitant wrinkling and buckle-delamination. This 2D model is able to predict the deformed compressed pattern based upon the mechanical properties of both film and substrate such as their thickness, critical stretch, elastic modulus and, Poisson's ratio

in the absence of pre-existing defects of any kind. PD results are compared against existing analytical solutions for the film and substrate interactions and a good agreement has been achieved. This study enables us to predict the occurrence of wrinkling and/or buckling and/or concomitant wrinkling and buckling for a given film and substrate when they are under compression.

- **Chapter 5:** In this chapter we formulate a nonlocal ordinary state based peridynamic formulation of plastic deformation based on the idea of mechanical sublayers. Each bond is regarded as being composed of a series of elasto-plastic subelements arranged in parallel. These subelements have a common modulus but different yield strains. Accordingly, as bond strain increases, the subelements yield successively, producing a piecewise linear approximation to uniaxial data which can be adjusted to achieve arbitrary accuracy by increasing the number of subelements. The model automatically incorporates the Bauschinger effect and models hysteretic cyclic loading. The proposed model is validated by comparison with the benchmark problems such as center-cracked tensile plate, and compact tension specimen.
- **Chapter 6:** Here, a new peridynamic model is proposed to model viscoplastic behavior of body armors against projectile ballistic impacts. The ultimate goal is to develop a parametric peridynamic model for simulation of ballistic impact in dual-material body armors. We propose two design patterns; one is composed of multidirectional fiber reinforced composites in the backing panel and ceramic-based front layers. The second design is a bio-inspired structure with superior ballistic impact resistance for wearable body armor systems. This is accomplished by replacing the conventional monolithic backing polymer by a multi-layered structure having multiple strong interfaces which deflect the stress waves generated by the penetrating projectile and, in turn, reduce the impact energy. The novelty of the proposed designs is the development of a reliable framework and computationally accurate and fast model for designing armor structures exhibiting unique ballistic impact resistance.
- **Chapter 7:** In this chapter we investigate the parallelization of PD algorithm as a particle-based method which is massively parallelizable. We have made a comprehensive comparison among three common parallelization systems: distributed and shared memory CPU systems using MPI and OpenMP, respectively, as well as GPU architecture using CUDA. We have built our parallel algorithms on a single benchmark problem where no damage is allowed and particles can move around different horizons. This is believed to be the most expensive computational method to solve PD problems. We have chosen this method intentionally to explore the efficiency and scalability of PD in the most rigorous situation. Results for weak and strong scaling efficiency are compared against each other and scaling performances for each model are evaluated and discussed. We show that these techniques can effectively speed up the PD simulations. We will conclude that CUDA implementation is shown to be the most powerful tool for massively parallel PD problems.

- **Chapter 8:** In this chapter we propose a new class of learnable predictive models using graph neural networks (GNN), which takes in materials properties and geometry and outputs their reaction in unseen environments under challenging loading conditions. In particular we investigate how GNNs can enhance PD theory by showing how by executing a limited number of PD simulations one can accurately learn dynamics of a physical system and generalize to new material structures. In this part of the dissertation, we take a key step towards revealing the benefits of combining machine learning based methods and continuum theory by exploiting compositional relations and power of statistical representation learning to open new paths for robust and efficient reasoning and decision-making.
- **Chapter 9:** We conclude with highlighting our contributions during the course of this PhD and shed some lights on future directions and possible extension of the current study.

Part I

Peridynamic Analysis of Materials  
Behavior

## Chapter 2

# Introduction to Peridynamic Theory

### 2.1 Preliminaries

Peridynamic theory assumes that a material is composed of finite number of particles that interact with each other via *bonds* within a specified limit, called the *horizon*. Every particle, interacts with a group of other particles, so called the *family members* within its horizon. Mechanical behavior of each particle *collectively* depends on its family members. Figure (2.1) shows two particles of the same family, located at  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in the reference configuration and their horizon limits when the motion has not yet started. Particles can undergo an arbitrary displacement, velocity, and/or force field externally which results in motion and deformation of bonds in the current configuration. Displacement of material points is denoted as

$$\mathbf{u}_i = \mathbf{y}_i - \mathbf{x}_i \quad (2.1)$$

where  $\mathbf{y}_i$  is the position of particle  $i$  in the current configuration. In addition to the particles' displacement, deformation also takes place at the bond level for all the bonds attached to the particle of interest (ex. particle  $i$ ) which forms a vector that stores the displacement of all  $N$  bonds. This is called the deformation *vector state* denoted as

$$\underline{\mathbf{Y}}(\mathbf{x}_i, t) \langle \mathbf{x}_j - \mathbf{x}_i \rangle = [(\mathbf{y}_1 - \mathbf{y}_i), (\mathbf{y}_2 - \mathbf{y}_i), \quad \cdots \quad (\mathbf{y}_N - \mathbf{y}_i)] \quad (2.2)$$

The *state* concept introduced by Silling (Silling, 2000) can be thought as a mapping function that *operates* on a single bond in the reference configuration and transfers it to the current configuration. The bond between particles  $i$  and  $j$  which has a length of  $|\mathbf{x}_i - \mathbf{x}_j|$  in the reference configuration can *stretch(or shorten)* upon deformation to the length of  $|\mathbf{y}_i - \mathbf{y}_j|$  to introduce the dimensionless *bond stretch* parameter which is analogous to *strain* in the classical theory and is the key concept in defining the *failure* of the material or generally the change of its behavior as an elastic, or plastic material. Stretch in the bond shared between particles  $i$  and  $j$  is defined as:

$$s_{ij} = \frac{|\mathbf{y}_j - \mathbf{y}_i| - |\mathbf{x}_j - \mathbf{x}_i|}{|\mathbf{y}_j - \mathbf{y}_i|} \quad (2.3)$$

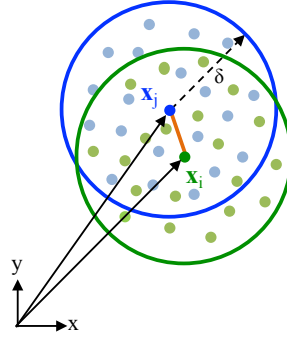


Figure 2.1: Family members of  $x_i$  and  $x_j$  centered at their horizons with radii of  $\delta$  share a bond in two-dimensional Cartesian space

Every particle is associated with a force density vector state which is the resultant of all forces exerted by its family members and is denoted as

$$\underline{\mathbf{T}}(\mathbf{x}_i, t) \langle \mathbf{x}_j - \mathbf{x}_i \rangle = [\mathbf{t}_{i1}, \mathbf{t}_{i2}, \quad \dots \quad \mathbf{t}_{iN}] \quad (2.4)$$

where  $\mathbf{t}_{ij}$  is the force exerted on particle  $i$  by particle  $j$  per unit volume. From the force vector prospective, direction and magnitude of vectors  $\mathbf{t}_{ij}$  and  $\mathbf{t}_{ji}$  divide the PD theory into three distinctive approaches which are illustrated in Fig. (2.2) as the bond-based theory by (Silling, 2000), the ordinary state-based theory by (Silling *et al.*, 2007), and the non-ordinary state-based theory by (Warren *et al.*, 2009).

As depicted in Fig. (2.2), in the bond-based framework, the force vectors between two family members are equal and parallel in opposite directions. This in fact, enforced a restriction in the theory to be only valid for materials with Poisson's ratio of  $1/3$  in  $2D$  and  $1/4$  in  $3D$ . Soon after, force state was introduced in ordinary state-based framework such that it was dependent upon the deformation of all its neighboring rather than pairwise interactions between particles. In ordinary stated-based approach  $\mathbf{t}_{ij}$  and  $\mathbf{t}_{ji}$  are no longer equal in magnitude but they are still parallel to each other. subsequently, the non-ordinary state-based framework was proposed as the most general PD framework in which  $\mathbf{t}_{ij}$  and  $\mathbf{t}_{ji}$  can have different magnitudes and directions allowing to use the constitutive equations of the classical theory. We are proposing our new plasticity model in the ordinary state based approach which can be also used in the bond-based framework.

Equation of motion in PD was first formulated by (Silling, 2000) which differs from its classical counterpart in the integral term instead of the divergence of the stress field which enables it to be applied anywhere in the material regardless of being a discontinuity such as cracks, voids, etc. PD equation of motion is given as

$$\rho(i) \ddot{\mathbf{u}}_i(\mathbf{x}_i, t) = \int_{\mathcal{H}} \{ \underline{\mathbf{T}}[\mathbf{x}_i, t] - \underline{\mathbf{T}}[\mathbf{x}_j, t] \} dV_{x_j} + \mathbf{b}(\mathbf{x}_i, t) \quad (2.5)$$



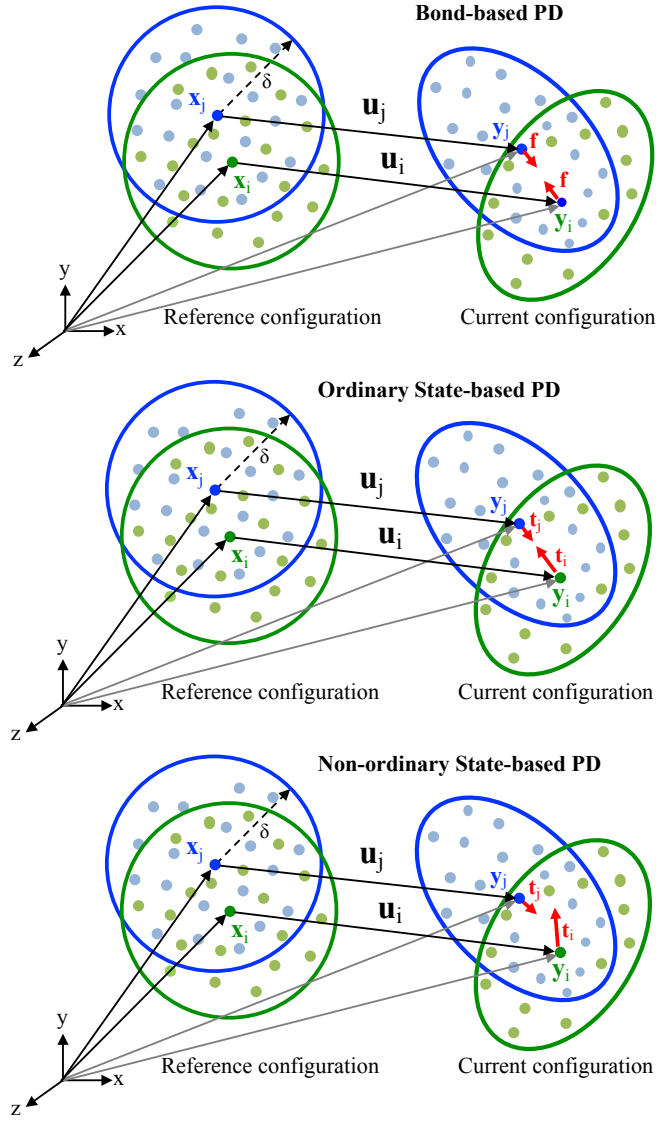


Figure 2.2: PD approaches: bond-based theory, ordinary state-based theory, non-ordinary state based theory

where  $\rho(\mathbf{x}_i)$  is the mass density of particle  $i$ ,  $\mathcal{H}$  confines the integral's bound over the horizon,  $dV$  is the incremental volume of each family members of  $i$ , and  $\mathbf{b}$  is the body forces exerted on particle  $i$ .

## 2.2 State-based Peridynamic Constitutive Equation

Due to the interaction between material points  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , a scalar-valued micropotential,  $w_{ij}$ , develops which depends on the material properties as well as the stretch between point

$\mathbf{x}_i$  and all other material points in its family.

$$w_{ij} = \hat{w}_{ij}(\mathbf{y}_{1^i} - \mathbf{y}_i, \mathbf{y}_{2^i} - \mathbf{y}_i, \dots) \quad (2.6)$$

$$w_{ji} = \hat{w}_{ji}(\mathbf{y}_{1^j} - \mathbf{y}_j, \mathbf{y}_{2^j} - \mathbf{y}_j, \dots) \quad (2.7)$$

$$w_{ij} \neq w_{ji} \quad (2.8)$$

The strain energy density,  $W_i$  of material point  $\mathbf{x}_i$  can be expressed as a summation of micropotentials,  $w_{ij}$ , arising from the interaction of material point  $\mathbf{x}_i$  and the other material points,  $\mathbf{x}_j$ , within its horizon in the form below: (Silling, 2000):

$$W_i = \frac{1}{2} \sum_{j=1}^{\infty} \frac{1}{2} (w_{ij} + w_{ji}) V_j \quad (2.9)$$

where  $V_j$  is the volume of particle  $j$

## Equation of motion

The PD equations of motion at material point  $\mathbf{x}_i$  can be derived by applying the principle of virtual work, i.e.,

$$\delta \int_{t_0}^{t_1} \underbrace{(K - U)}_L dt = 0 \quad (2.10)$$

$K$  and  $U$  represent kinematic and potential energies in the system. Principle of virtual work is met by solving the Lagrange's equation :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{u}_i} \right) - \frac{\partial L}{\partial u_i} = 0 \quad (2.11)$$

Total kinetic energy:

$$K = \sum_{i=1}^{\infty} \frac{1}{2} \rho_i \dot{u}_i \cdot \dot{u}_i V_i \quad (2.12)$$

Total potential energy:

$$U = \sum_{i=1}^{\infty} W_i V_i - \sum_{i=1}^{\infty} (b_i \cdot u_i) V_i = \sum_{i=1}^{\infty} \frac{1}{2} \sum_{j=1}^{\infty} \frac{1}{2} (w_{ij} + w_{ji}) V_j - b_i \cdot u_i V_i \quad (2.13)$$

Replacing Eqs. 2.12 and 2.13 into 2.11 will results in:

$$L = K - U = \sum_{i=1}^{\infty} \frac{1}{2} \rho_i \dot{u}_i \cdot \dot{u}_i V_i - \left( \sum_{i=1}^{\infty} W_i V_i - \sum_{i=1}^{\infty} (b_i \cdot u_i) V_i \right) \quad (2.14)$$

$$= \sum_{i=1}^{\infty} \frac{1}{2} \rho_i \dot{u}_i \cdot \dot{u}_i V_i - \sum_{i=1}^{\infty} \frac{1}{2} \sum_{j=1}^{\infty} \frac{1}{2} (w_{ij} + w_{ji}) V_j + b_i \cdot u_i V_i \quad (2.15)$$

Plugging back to the Lagrange's equation and using chain rule:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{u}_i}\right) - \frac{\partial L}{\partial u_i} = 0$$

$$\rho_i \ddot{u}_i V_i + \sum_{j=1}^{\infty} \frac{1}{2} \left( \sum_{k=1}^{\infty} \frac{\partial w_{ij}}{\partial (\mathbf{y}_j - \mathbf{y}_i)} V_k \right) \frac{\partial (\mathbf{y}_j - \mathbf{y}_i)}{\partial \mathbf{u}_i} + \sum_{j=1}^{\infty} \frac{1}{2} \left( \sum_{k=1}^{\infty} \frac{\partial w_{ji}}{\partial (\mathbf{y}_i - \mathbf{y}_j)} V_k \right) \frac{\partial (\mathbf{y}_i - \mathbf{y}_j)}{\partial \mathbf{u}_i} - \mathbf{b}_i \Big) V_i = 0 \quad (2.16)$$

$$\left. \begin{array}{l} \mathbf{y}_i = \mathbf{x}_i + \mathbf{u}_i \\ \mathbf{y}_j = \mathbf{x}_j + \mathbf{u}_j \end{array} \right\} \Rightarrow \frac{\partial (\mathbf{y}_i - \mathbf{y}_j)}{\partial \mathbf{u}_i} = I \quad (2.17)$$

$$\frac{\partial (\mathbf{y}_j - \mathbf{y}_i)}{\partial \mathbf{u}_i} = -I$$

$$\rho_i \ddot{u}_i = \sum_{j=1}^{\infty} \frac{1}{2} \left( \sum_{k=1}^{\infty} \frac{\partial w_{ik}}{\partial (\mathbf{y}_j - \mathbf{y}_i)} V_k \right) - \sum_{j=1}^{\infty} \frac{1}{2} \left( \sum_{k=1}^{\infty} \frac{\partial w_{ki}}{\partial (\mathbf{y}_i - \mathbf{y}_j)} V_k \right) + \mathbf{b}_i \quad (2.18)$$

Force state:

$$\mathbf{t}_{ij}(\mathbf{u}_j - \mathbf{u}_i, \mathbf{x}_j - \mathbf{x}_i, t) = \frac{1}{2V_j} \sum_{k=1}^{\infty} \frac{\partial w_{ik}}{\partial (\mathbf{y}_j - \mathbf{y}_i)} V_k = \underline{\mathbf{T}}(\mathbf{x}_i, t) \langle \mathbf{x}_j - \mathbf{x}_i \rangle \quad (2.19)$$

$$\mathbf{t}_{ji}(\mathbf{u}_i - \mathbf{u}_j, \mathbf{x}_i - \mathbf{x}_j, t) = \frac{1}{2V_j} \sum_{k=1}^{\infty} \frac{\partial w_{ki}}{\partial (\mathbf{y}_i - \mathbf{y}_j)} V_k = \underline{\mathbf{T}}(\mathbf{x}_j, t) \langle \mathbf{x}_i - \mathbf{x}_j \rangle \quad (2.20)$$

The force state consists of the constitutive equation of the material in PD. (Oterkus & Madenci, 2012) derived the PD constitutive equation as follows. They first considered PD equation of motion for local interactions (traction vectors) and matched the terms with its equivalent form in the classical theory which resulted in a relation between Cauchy stress components and PD force states. By replacing the force states with Cauchy stress components in the well-established strain energy density function of the classical theory and comparing the corresponding terms, they found the strain energy density function in peridynamics as a function of force state as below

$$W_i = a_1 \theta_i^2 - a_2 \theta_i \Delta T_i + a_3 \Delta T_i^2 + b \sum_{j=1}^N \omega_{ij} \left( (|\mathbf{y}_j - \mathbf{y}_i| - |\mathbf{x}_j - \mathbf{x}_i|) - \alpha \Delta T_i |\mathbf{x}_j - \mathbf{x}_i| \right)^2 V_j \quad (2.21)$$

where  $a_1$ ,  $a_2$ ,  $a_3$ , and  $b$  are PD constants which are defined in Table 2.1.  $\Delta T$  is the temperature change in the material,  $\alpha$  is the coefficient of thermal expansion of the material,  $\omega_{ij}$  is the influence function between particle  $i$  and  $j$  denoted as below and shown in Fig. [ref:

$$\omega_{ij} = \frac{\delta}{|\mathbf{x}_j - \mathbf{x}_i|} \quad (2.22)$$

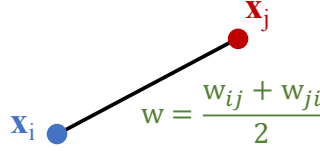


Table 2.1: Peridynamic material constants

| Dimension | Peridynamic parameters                       |                                |                                                      |                                  |                               |
|-----------|----------------------------------------------|--------------------------------|------------------------------------------------------|----------------------------------|-------------------------------|
| 3D        | $a_1 = \frac{1}{2}(\kappa - \frac{5\mu}{3})$ | $a_2 = \alpha(3\kappa - 5\mu)$ | $a_3 = \frac{9}{2}\alpha^2(\kappa - \frac{5\mu}{3})$ | $b = \frac{15\mu}{2\pi\delta^5}$ | $d = \frac{9}{4\pi\delta^4}$  |
| 2D        | $a_1 = \frac{1}{2}(\kappa - 2\mu)$           | $a_2 = 2\alpha(\kappa - \mu)$  | $a_3 = 2\alpha^2(\kappa - \mu)$                      | $b = \frac{6\mu}{\pi h\delta^4}$ | $d = \frac{2}{\pi h\delta^3}$ |
| 1D        | $a_1 = 0$                                    | $a_2 = 0$                      | $a_3 = 0$                                            | $b = \frac{E}{2A\delta^3}$       | $d = \frac{1}{2A\delta^2}$    |

and  $\theta$  is the dilation denoted as below

$$\theta_i = d \sum_{j=1}^N \omega_{ij}(s_{ij} - \alpha T_i) \frac{\mathbf{y}_j - \mathbf{y}_i}{|\mathbf{y}_j - \mathbf{y}_i|} \cdot (\mathbf{x}_j - \mathbf{x}_i) V_j + 3\alpha \Delta T_i \quad (2.23)$$

According to (Silling, 2000), force density of a bond is the derivative of the strain energy density function with respect to the relative displacements of particles sharing that bond

$$\mathbf{t}_{ij} = \frac{1}{V_j} \frac{\partial W_i}{\partial (|\mathbf{y}_j - \mathbf{y}_i|)} \frac{\mathbf{y}_j - \mathbf{y}_i}{|\mathbf{y}_j - \mathbf{y}_i|} \quad (2.24)$$

By taking derivative of Eq. (2.21), force state density can be derived as

$$\mathbf{t}_{ij} = 2\delta \left( d \frac{\cos[\angle(\mathbf{y}_j - \mathbf{y}_i, \mathbf{x} - \mathbf{x}_i)]}{|\mathbf{x}_j - \mathbf{x}_i|} (a_1 \theta_i - \frac{1}{2} a_2 \Delta T_i) + b(s_{ij} - \alpha \Delta T_i) \right) \frac{\mathbf{y}_j - \mathbf{y}_i}{|\mathbf{y}_j - \mathbf{y}_i|} \quad (2.25)$$

where

$$\cos[\angle(\mathbf{y}_j - \mathbf{y}_i, \mathbf{x} - \mathbf{x}_i)] = \frac{\mathbf{y}_j - \mathbf{y}_i}{|\mathbf{y}_j - \mathbf{y}_i|} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \quad (2.26)$$

Table 2.1 shows the PD material constant in one, two, and three dimensional framework:

Several methods have been proposed to evaluate a PD equivalent for the path-independent J-integral which is originally proposed by (Rice, 1968) when plastic deformation occurs in the material. Initially, (Silling & Lehoucq, 2010) derived a PD version of the J-integral that computes the energy dissipation rate of moving discontinuities. (Foster *et al.*, 2011) computed J-integral via an energy approach for viscoelastic materials using non-ordinary state based PD. (Hu *et al.*, 2012b) used the crack infinitesimal virtual extension approach to derive an algorithm for J-integral calculation in bond-based peridynamics. (Madenci & Oterkus, 2016) used an energy based framework to compute J-integral by calculating the

total work required to eliminate the bonds across the discontinuity surface. We used the latter formulation because it requires the least amount of computations in the state based formulation we are using developed by the same authors (Madenci & Oterkus, 2014). They showed that the strain energy required to eliminate a PD bond between two particles of the same family but located on opposite sides of a crack surface can be obtained as

$$W_{ij} = \frac{1}{2} \frac{w_{ij} + w_{ji}}{2} V_j V_i \quad (2.27)$$

where  $w_{ij}$  is called the elastic-plastic micropotential or the area under the force density vs. stretch  $(t_{ij} - s_{ij})$  per bond and is defined as

$$w_{ij} = \int_0^{s_{ij}} t_{ij} |\mathbf{x}_j - \mathbf{x}_i| ds_{ij} \quad (2.28)$$

Therefore, the J-integral or the total strain energy required to eliminate all the PD interactions across the unit crack surface  $A$  is

$$J = \frac{1}{2A} \sum_{\mathcal{H}_i} \sum_{\mathcal{H}_j} (w_{ij} + w_{ji}) V_i V_j \quad (2.29)$$

where  $\mathcal{H}_i$  and  $\mathcal{H}_j$  denote the family members of particle  $i$  and  $j$  that are on the same side of the crack, respectively and  $A = h\Delta x$  with  $h$  being the thickness.

## 2.3 Numerical solution scheme

Peridynamics is mesh-free meaning that the domain should be discretized into nodes to form a grid that has no geometrical connections (Silling & Askari, 2005). A uniform grid is used here with a horizon of  $\delta = 3.015 \Delta x$  which is a common choice in PD simulations. Each node has a volume of  $\Delta x^3$ . The equation of motion at the  $n^{th}$  time-step can be discretized as below

$$\rho(\mathbf{x}_i) \ddot{\mathbf{u}}_i(\mathbf{x}_i, t) = \sum_{j=1}^{N_{\mathcal{H}}} (\mathbf{t}_{ij}^n - \mathbf{t}_{ji}^n) \nu_j V_j + \mathbf{b}(\mathbf{x}_i, t) \quad (2.30)$$

where  $\nu_j$  is the area correction factor adapted from (Madenci & Oterkus, 2014).

Time integration is carried out explicitly using forward difference to update the velocity with

$$\dot{\mathbf{u}}_i^{n+1} = \ddot{\mathbf{u}}_i^n \Delta t + \dot{\mathbf{u}}_i^n \quad (2.31)$$

Displacements are updated using backward differences:

$$\mathbf{u}_i^{n+1} = \dot{\mathbf{u}}_i^{n+1} \Delta t + \mathbf{u}_i^n \quad (2.32)$$

Since we are using an explicit method, stability of the integration should be checked by the following equation which is thoroughly derived and explained in (Oterkus & Madenci, 2012; Madenci & Oterkus, 2014)

$$\Delta t < \sqrt{\frac{2\rho(\mathbf{x}_i)}{\sum_{j=1}^N \left[ \left( \sum_{m=1}^N 2|\alpha_{ijm}|V_m + \sum_{n=1}^N 2|\gamma_{ijn}|V_n \right) + \beta_{ij} \right] V_j}} \quad (2.33)$$

where

$$\beta_{ij} = \frac{4b\delta}{|\mathbf{x}_j - \mathbf{x}_i|}, \quad \alpha_{ijm} = \frac{2ad^2\delta^2}{|\mathbf{x}_j - \mathbf{x}_i| |\mathbf{x}_m - \mathbf{x}_i|}, \quad \gamma_{ijn} = \frac{2ad^2\delta^2}{|\mathbf{x}_j - \mathbf{x}_i|} |\mathbf{x}_n - \mathbf{x}_i| \quad (2.34)$$

Last thing needed is the initial condition for the above integration equations for both velocity and displacement.

## 2.4 Peridynamic pseudocode

A pseudocode that describes the PD algorithm is shown in Algorithm 1.

---

### Algorithm 1 PD ALGORITHM

---

- 1: Given:  $\delta$ ,  $\Delta t$ , Bulk modulus ( $K$ ), Shear modulus ( $G$ ), Critical stretch ( $s_{cr}$ )
  - 2: Initialize particles coordinates
  - 3: Initialize displacements and velocities
  - 4: **for** each time step  $t = 1$  to  $T$  **do**
  - 5:   **for** each particle  $i = 1$  to  $n$  **do**
  - 6:     **for** each particle  $j = 1$  of particle  $i$ 's family **do**
  - 7:        $\xi \leftarrow \mathbf{x}_j - \mathbf{x}_i$
  - 8:        $\eta \leftarrow \mathbf{u}_j - \mathbf{u}_i$
  - 9:       update  $scr$ ,  $fac$
  - 10:        $\mathbf{F}_i \leftarrow \mathbf{F}_i + bc \eta \frac{\eta - \xi}{\xi \Delta x^3 scr fac (\mathbf{x}_j + \mathbf{u}_j - \mathbf{x}_i - \mathbf{u}_i)}$
  - 11:       **if** Bond stretch  $> s_{cr}$  :
  - 12:          Break the  $j$ 's bond with  $i$
  - 13:   **for** each particle  $i = 1$  to  $n$  **do**
  - 14:      $\mathbf{u}_i, \mathbf{v}_i \leftarrow \text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$
  - 1: **procedure**  $\text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$
  - 2:    $\mathbf{a}_i \leftarrow \mathbf{F}_i / \rho$
  - 3:    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{a}_i \Delta t$
  - 4:    $\mathbf{u}_i \leftarrow \mathbf{u}_i + \mathbf{v}_i \Delta t$
  - 5:   **return**  $\mathbf{u}_i, \mathbf{v}_i$
-

## 2.5 Original Formulation

As described in chapter 1, there exists two different formulations in PD literature. What we have discussed so far in this chapter has been devoted to the notation introduced in (Oterkus & Madenci, 2012). However, PD was introduced with a different notation by (Silling, 2000). For the sake of completeness, we briefly show how different this notation is from the rest of what has been used in this thesis.

Bond-based peridynamics presumes the existence of a pair-wise force function between any two particles, which is independent of the deformation associated with other particles (Silling, 2000) and has been developed for a Poisson's ratio of 0.33 and 0.25 for 2D and three-dimensional (3D) problems, respectively. State-based peridynamics is based on a more general theory, which uses a more comprehensive constitutive model derived based on force- and deformation-state concepts (Silling *et al.*, 2007). To obtain the force state at each particle, the deformation (stretching) of all bonds within the horizon of each particle are considered without assuming a specific value of the Poisson's ratio. Similarities between state-based peridynamics and continuum theory have been reported (Silling *et al.*, 2007; Lehoucq & Silling, 2008), including the convergence of state-based peridynamics to classical elasticity theory (Silling & Lehoucq, 2008).

The general 3D peridynamics equation of motion is given by (Silling *et al.*, 2007)

$$\rho(\mathbf{x}_i)\ddot{\mathbf{u}}(\mathbf{x}_i, t) = \int_{\mathcal{H}} \left( \underline{\mathbf{T}}[\mathbf{x}_i, t] \langle \mathbf{x}_j - \mathbf{x}_i \rangle - \underline{\mathbf{T}}[\mathbf{x}_j, t] \langle \mathbf{x}_i - \mathbf{x}_j \rangle \right) dV_j + \mathbf{b}(\mathbf{x}_i, t) \quad (2.35)$$

where  $\rho$  is the mass density,  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are the position vectors of particles  $i$  and  $j$ , respectively,  $\mathbf{u}$  is the displacement field,  $\mathcal{H}$  is the domain of the spherical horizon with a radius  $\delta$ ,  $\underline{\mathbf{T}}$  is the force vector state field,  $\mathbf{b}$  is the body force density field,  $t$  is the time, and  $dV_j$  is the volume of particle  $j$ . In the present analysis, the deformable materials are assumed to be ordinary, implying that the force between two particles acts in the bond direction.

For ordinary materials, the force vector is given by (Silling *et al.*, 2007)

$$\underline{\mathbf{T}} = \underline{t} \underline{\mathbf{M}} \quad (2.36)$$

where  $\underline{t}$  is the scalar force state and  $\underline{\mathbf{M}}$  is the deformation direction vector. In the linear peridynamics solid (LPS) model, the force scalar state is defined by (Silling *et al.*, 2007)

$$\underline{t} = \frac{3K\theta[\mathbf{x}, t]}{m[\mathbf{x}]} \underline{\omega} \langle \underline{\boldsymbol{\xi}} \rangle \underline{x} \langle \underline{\boldsymbol{\xi}} \rangle + \frac{15G}{m[\mathbf{x}]} \underline{\omega} \langle \underline{\boldsymbol{\xi}} \rangle \underline{e}^d[\mathbf{x}, t] \quad (2.37)$$

where  $K$  and  $G$  are the bulk and shear modulus, respectively,  $\theta$  is the dilatation,  $m$  is the weighted volume,  $\underline{e}^d$  is the deviatoric component of the extension scalar state  $\underline{e}$ , and  $\underline{\omega}$  is the

influence function. These parameters can be defined as following (Silling *et al.*, 2007):

$$\theta[\mathbf{x}, t] = \frac{3}{m[\mathbf{x}]} \int_{\mathcal{H}} \underline{\omega}(\boldsymbol{\xi}) \underline{x}(\boldsymbol{\xi}) \underline{e}[\mathbf{x}, t](\boldsymbol{\xi}) dV \quad (2.38)$$

$$m[\mathbf{x}] = \int_{\mathcal{H}} \underline{\omega}(\boldsymbol{\xi}) \underline{x}(\boldsymbol{\xi}) \underline{x}(\boldsymbol{\xi}) dV \quad (2.39)$$

$$\underline{e}[\mathbf{x}, t](\boldsymbol{\xi}) = \|\boldsymbol{\xi} + \boldsymbol{\eta}\| - \|\boldsymbol{\xi}\| \quad (2.40)$$

$$\underline{e}^d[\mathbf{x}, t](\boldsymbol{\xi}) = \underline{e}[\mathbf{x}, t](\boldsymbol{\xi}) - \underline{e}^i[\mathbf{x}, t](\boldsymbol{\xi}) = \underline{e}[\mathbf{x}, t](\boldsymbol{\xi}) - \frac{1}{3}\theta[\mathbf{x}, t] \underline{x}(\boldsymbol{\xi}) \quad (2.41)$$

where  $\boldsymbol{\xi} = \mathbf{x}_j - \mathbf{x}_i$  is the relative position vector between particles  $i$  and  $j$  in the reference configuration and  $\boldsymbol{\eta} = \mathbf{u}(\mathbf{x}_j, t) - \mathbf{u}(\mathbf{x}_i, t)$  is the relative displacement vector between particles  $i$  and  $j$  at time  $t$ .

Because of the highly disordered structure of  $a$ -C films (Charitidis, 2010), they can be modeled as isotropic materials with an influence function  $\underline{\omega}(\boldsymbol{\xi}) = 1/\|\boldsymbol{\xi}\|$ , as suggested elsewhere (Parks *et al.*, Sandia Report 2011-8523, 2010).

Damage is assumed to occur when bond stretching exceeds a predefined critical stretch  $s_c$ , given by (Silling & Askari, 2005; Ha & Bobaru, 2011)

$$s_c = \sqrt{\frac{4\pi G_I}{9E\delta}} \quad (2.42)$$

where  $G_I$  is the critical energy release rate corresponding to the mode I stress intensity factor  $K_I$  (i.e.,  $G_I = K_I^2/E'$ , where  $E' = E$  (plane stress) or  $E/(1 - \nu^2)$  (plane strain)). Eq. (2.42) indicates that  $s_c$  is a function of the material properties and characteristic length scale of the analyzed body, i.e., the horizon radius  $\delta$ .

Damage at a given material point (particle) is defined as the ratio of the number of broken bonds to the total number of bonds  $D$ . Because  $D$  assumes values between 0 (no damage) and 1 (full damage) (Silling & Askari, 2005), it can be used as a damage index to characterize the extent of material removal ( $D = 1$ ) and the evolution of permanent damage ( $0 < D < 1$ ) in the wear model.

### 2.5.1 Body discretization and computational details

To obtain a solution for the 2D version of Eq. (2.35), the body is discretized by a uniform grid ( $\Delta x = \Delta y$ ) and the integral is replaced by a summation including all interacting particles within the horizon of a given particle. Thus, Eq. (2.35) can be expressed as

$$\rho_i \ddot{\mathbf{u}}_i^n = \sum_{j=1}^{N_{\mathcal{H}}} \mathcal{F}(\mathbf{x}_i^n, \mathbf{x}_j^n, \mathbf{x}_i^{n-1}, \mathbf{x}_j^{n-1}) V_j + \mathbf{b}_i^n \quad (2.43)$$

where the superscripts denote the time step and  $N_{\mathcal{H}}$  is the total number of particles interacting with the particle of interest within its horizon. Time integration of Eq. (2.43) using



the central difference method yields the position and velocity of each particle at time step  $(n + 1)$ . The nodal area of the particles laying on the horizon boundaries is accordingly modified (Parks *et al.*, 2008).

In addition to the force vector state obtained from Eq. (2.36), short-range forces are also included in the present analysis by introducing a short-range particle interaction distance  $d_{pi} = \min \{0.9 \|\mathbf{x}_p - \mathbf{x}_i\|, 1.35(r_p + r_i)\}$ , where  $\mathbf{x}_p$  and  $r_p$  are the position and radius of particle  $p$  in the vicinity of particle  $i$ , respectively, and  $r_i$  is the radius of particle  $i$ , which is set equal to one-half of the grid size (i.e.,  $r_i = \Delta x/2$ ) (Parks *et al.*, Sandia Report 2011-8523, 2010).

## Chapter 3

# Studying Wear and Friction in Carbon Thin Films with Peridynamic

### 3.1 Amorphous carbon thin films

Thin films are used as protective overcoats in a wide range of applications where the tribological properties of proximal surfaces are of paramount importance to the functionality and endurance of mechanical components possessing contact interfaces. For example, thin films of amorphous carbon (*a*-C) play a critical role in the reliability and performance of magnetic recording devices because they protect the magnetic head and hard disk surfaces against mechanical wear during intermittent contact and inhibit corrosion of the magnetic medium of the hard disk. Because of the extremely small *a*-C film thickness and the occurrence of head-disk surface interactions at nanoscopic surface protrusions (asperities), knowledge of the nanoscale tribological and mechanical properties of thin *a*-C films is of high technological importance.

The nano-mechanical/tribological properties of *a*-C films are greatly affected by the type of carbon atom hybridization and the hydrogen content. Other elements (e.g., Si, N, B, F, and O) can be added to modify the electromechanical properties of *a*-C films (Charitidis, 2010). The structure and elemental content of *a*-C films strongly depend on the intricacies of the deposition process, which controls film nucleation and growth (Charitidis, 2010; Lifshitz, 1996; Grill, 1999). Thus, small variations in the deposition conditions may result in vastly different film properties. In view of the time consuming experimental techniques available for nanoscale mechanical and tribological testing of thin films, alternative approaches must be used to examine the effects of structural changes on resulting film properties.

High contents of tetrahedral carbon atom hybridization ( $sp^3$ ) characterize the structure of *a*-C films exhibiting diamond-like behavior, whereas high contents of trigonal carbon atom hybridization ( $sp^2$ ) generally produce graphitic-like film behavior. A continuum description does not account for local differences in nanostructure (Luan & Robbins, 2005), whereas molecular dynamics (MD) is limited by high computational cost, model size, and type of

potential function used to describe atomic interaction (Alder & Wainwright, 1959). Therefore, nonlocal computational approaches, which are not subjected to the aforementioned restrictions, must be developed to enhance the study of the interdependence of structure and material behavior at the nanoscale.

Peridynamics (Silling, 2000) is a relatively new theory, which promises to bridge the material gap in computational mechanics. Peridynamics is a continuum version of MD, which uses integral equations of motion to offset complexities associated with material discontinuities (e.g., defects, edges, and sharp corners) instead of the conventional partial differential equations used in classical mechanics and does not rely on *a priori* assumed defect or damage criteria (e.g., crack growth direction). Because of the mathematical simplicity and computational affordability, peridynamics has been used to analyze various computationally intense problems, such as dynamic fracture in brittle (Ha & Bobaru, 2011; Bobaru & Hu, 2012; Liu & Hong, 2012; Lipton, 2014; Ha & Bobaru, 2010) and composite (Askari *et al.*, 2006; Xu *et al.*, 2008; Kilic *et al.*, 2009; Hu *et al.*, 2011; Hu *et al.*, 2012a) materials, multi-scale damage (Askari *et al.*, 2008; Alali & Lipton, 2012), and damage of nanofiber networks, including long-range effects of van der Waals forces on nanofiber deformation (Bobaru, 2007; Bobaru & Silling, 2004; Bobaru *et al.*, 2005; Silling & Bobaru, 2005). Moreover, peridynamics has been used in failure analyses dealing with thin film cracking in electronic packaging (Agwai *et al.*, 2008; Agwai *et al.*, 2011; Agwai *et al.*, 2009) and also in conjunction with atomic force microscopy and nanoindentation techniques to determine the mechanical properties of ultrathin films (Celik *et al.*, 2009).

The objective of this study is to introduce a two-dimensional (2D) peridynamics analysis of the nanotribological behavior of thin *a*-C films. Simulation results of the coefficient of friction and depth of wear track due to a rigid (diamond) tip sliding against *a*-C films of different thickness and nanomechanical properties are presented and compared with experimental results of a previous study (Lu & Komvopoulos, 2001) to validate the accuracy of the developed peridynamics models.

## 3.2 Peridynamics friction and wear models

State-based peridynamics friction and wear models are presented in this section and simulation results are compared with published experimental results of the nanoscale tribological properties of thin *a*-C films (Lu & Komvopoulos, 2001) to illustrate the validity of the developed models. A 2D analysis of the sliding process is valid provided the depth of penetration is significantly less than the width of the resulting plowing (wear) track (Komvopoulos *et al.*, 1985).

Long-range forces may also have a strong effect on nanoscale deformation and, despite the continuum nature of peridynamic, it is possible to incorporate potential force functions from MD analysis in the force state of peridynamic (Bobaru, 2007; Silling & Bobaru, 2005; Bobaru *et al.*, 2005). However, for a separation distance of 2 nm, long-range forces reach

Table 3.1: Mechanical properties and critical stretch of Si substrate. <sup>(a)</sup>Ref. (Lu & Komvopoulos, 2001) and <sup>(b)</sup>Ref. (Agwai *et al.*, 2011)

| Material | Elastic modulus <sup>(a)</sup> (GPa) | Poisson's ratio <sup>(a)</sup> | Density <sup>(a)</sup> (g/cm <sup>3</sup> ) | Critical stretch <sup>(b)</sup> |
|----------|--------------------------------------|--------------------------------|---------------------------------------------|---------------------------------|
| Silicon  | 132                                  | 0.278                          | 2.329                                       | 0.01                            |

$\sim 10\%$  of their peak values (Bobaru, 2007). Consequently, because the grid size used in the present analysis is less than 2 nm, long-range forces are not considered for simplicity.

Since in all simulation cases the ratio of the wear depth to the wear track width is less than 0.1, a 2D peridynamics analysis of the sliding friction and wear processes is justifiable. All simulations were performed with a custom-made peridynamics code written in Fortran 90/95 and executed on a Linux platform with a quad-core 2.33 GHz Intel Xeon E5345 CPU.

### 3.2.1 Friction model

Fig. 3.1 schematically shows a rigid spherical tip of radius  $R$  under normal load  $P$ , which is sliding against a thin  $a$ -C film firmly attached to a thick Si substrate. Because of the high elastic modulus of diamond, in all numerical simulations the tip is modeled as rigid. The center of the tip is initially set at a distance equal to  $R - \Delta x/2$  from the film surface. Short-range forces inhibit the development of particle-particle distances less than  $d_{pi}$ , defined in section 3. Both film and substrate materials are assumed to be isotropic, predominantly exhibiting brittle behavior. The elastic properties, density, and critical stretch of the Si substrate are given in Table 3.1, whereas the thickness, root-mean-square (rms) roughness, elastic properties, and density of all  $a$ -C films examined in this study are given in Table 6.1. The film density was calculated from the relation  $\rho = 1.37 + E^{2/3}/44.65$ , where  $\rho$  and  $E$  are given in g/cm<sup>3</sup> and GPa, respectively (Casiraghi *et al.*, 2007). The number of particles used to discretize the film and substrate media in each friction simulation are also given in Table 6.1. In all friction simulations, the tip radius is equal to 20  $\mu\text{m}$ .

To enhance the convergence, load-control sliding experiments were simulated by the following method. First, the normal load was incrementally applied using several time steps until the desired load (in the range of 50-400  $\mu\text{N}$ ) was reached. This incremental loading procedure is similar (though faster) to that used in the experimental study (Lu & Komvopoulos, 2001). Subsequently, the tip was traversed in the  $x$ -direction at a constant velocity  $V = 0.4 \mu\text{m/s}$  by a total distance  $S = 4 \mu\text{m}$ . To avoid boundary effects on the friction results, the distance of the left and right boundaries of the discretized domain from the initial ( $x/S = 0$ ) and final ( $x/S = 1$ ) tip positions was set equal to  $S/2$  (Fig. 1). An adaptive dynamic relaxation (ADR) method similar to that presented in (Kilic & Madenci, 2010), which was accordingly modified for state-based formulation, was used in the friction analysis. The time step in the ADR analysis of friction was set equal to 0.01 s. Artificial damping was used in the equations of particle and tip motion. For the calculation of the damping coefficient of the rigid tip, the stiffness was increased by a factor of 10 to account for the rigidity of the

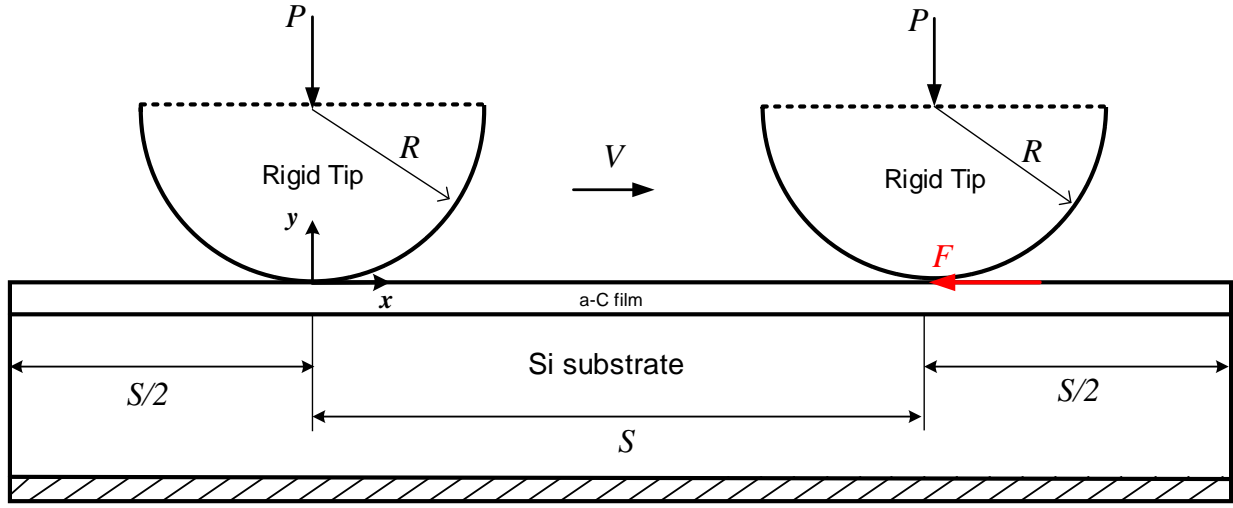


Figure 3.1: Schematic of peridynamics friction model of a spherical diamond (rigid) tip sliding at constant velocity  $V$  against a thin  $a$ -C film, which is firmly adhered to a thick Si substrate. The tip slides from left ( $x/S = 0$ ) to right ( $x/S = 1$ ) by a total distance  $S$ . The shaded layer at the bottom of the substrate is modeled as rigid. The coefficient of friction is obtained as the ratio of the computed tangential (friction) force  $F$ , which opposes tip sliding, and the applied normal load  $P$ . The film thickness and the tip radius are not drawn to scale.

tip. Similar to the friction experiments reported in (Lu & Komvopoulos, 2001), only elastic deformation is modeled in the friction simulations, i.e., irreversible damage such as bond breakage is not included in the friction model. The initial boundary conditions used for time integration are zero displacements and velocities in all directions at all particles and the tip center. Films of thickness larger than 10 nm are examined because the grid size is less than 2 nm. In addition, because the rms roughness of the films (in the range of 0.15–0.51 nm (Lu & Komvopoulos, 2001)) is significantly smaller than the grid size, both film and substrate media are modeled as perfectly smooth.

### 3.2.2 Wear model

Fig. 3.2 shows a schematic of the wear model consisting of a sharp rigid (diamond) conospherical probe with a tip radius  $R$  plowing through an  $a$ -C film, which is firmly attached to a thick Si substrate. After incremental loading of the probe to the desired normal load  $P = 10 \mu\text{N}$  (load-control simulations) at  $x/S = 0$ , the probe was traversed in the  $x$ -direction at a constant velocity  $V = 4 \mu\text{m/s}$  by a total distance  $S = 1 \mu\text{m}$  and was finally unloaded at  $x/S = 1$ . To avoid boundary effects on the wear results, the distance of the left and right boundaries of the discretized domain from the initial ( $x/S = 0$ ) and final ( $x/S = 1$ ) tip positions was set equal to  $S/4$ . The thickness, rms roughness, elastic properties, density, and critical stretch of  $a$ -C films used in the wear model and the number of particles used to

Table 3.2: Thickness, roughness, elastic properties, and density of *a*-C films and numbers of film and substrate particles used in the peridynamics friction analysis (Lu & Komvopoulos, 2001).

| Film No. | Thickness<br>(nm) | Roughness,<br>rms<br>(nm) | Elastic<br>modulus<br>(GPa) | Poisson's<br>ratio | Density<br>(g/cm <sup>3</sup> ) | Number of particles |            |
|----------|-------------------|---------------------------|-----------------------------|--------------------|---------------------------------|---------------------|------------|
|          |                   |                           |                             |                    |                                 | Film                | Substrate  |
| 1        | 31                | 0.51                      | 105                         | 0.278              | 3.139                           | 5000 × 19           | 5000 × 606 |
| 2        | 34                | 0.20                      | 197                         | 0.278              | 4.058                           | 5000 × 21           | 5000 × 604 |
| 3        | 39                | 0.15                      | 206                         | 0.278              | 4.143                           | 5000 × 24           | 5000 × 601 |
| 4        | 53                | 0.27                      | 139                         | 0.278              | 3.500                           | 5000 × 33           | 5000 × 592 |
| 5        | 59                | 0.23                      | 101                         | 0.278              | 3.094                           | 5000 × 36           | 5000 × 589 |
| 6        | 69                | 0.15                      | 192                         | 0.278              | 4.017                           | 5000 × 43           | 5000 × 582 |
| 7        | 95                | 0.24                      | 155                         | 0.278              | 3.661                           | 5000 × 59           | 5000 × 566 |

Table 3.3: Thickness, roughness, elastic properties, density, and critical stretch of *a*-C films and numbers of film and substrate particles used in the peridynamics wear analysis (Lu & Komvopoulos, 2001).

| Film No. | Thickness<br>(nm) | Roughness,<br>rms<br>(nm) | Elastic<br>modulus<br>(GPa) | Poisson's<br>ratio | Density<br>(g/cm <sup>3</sup> ) | Critical<br>stretch | Number of particles |           |
|----------|-------------------|---------------------------|-----------------------------|--------------------|---------------------------------|---------------------|---------------------|-----------|
|          |                   |                           |                             |                    |                                 |                     | Film                | Substrate |
| 8        | 17                | 0.19                      | 113                         | 0.278              | 3.230                           | 0.0125              | 937 × 10            | 937 × 615 |
| 9        | 22                | 0.18                      | 203                         | 0.278              | 4.115                           | 0.0125              | 937 × 13            | 937 × 612 |
| 10       | 10                | 0.2                       | 226                         | 0.278              | 4.317                           | 0.0125              | 937 × 6             | 937 × 619 |

discretize the film and the substrate in the wear model are given in Table 3.3. In all wear simulations, the probe tip radius is equal to 1  $\mu\text{m}$  and the initial displacements and velocities of the tip and all particles are set equal to zero. The previously mentioned ADR technique (Kilic & Madenci, 2010) with a time step of 0.001 s was also used in the wear analysis.

In the wear simulations, irreversible damage in the wake of the plowing tip comprises bond breakage. Therefore, a critical bond stretch was used to capture bond breakage. In addition to the critical bond stretch of the substrate (Table 1) and film (Table 3.3) materials, a conservative estimate of the critical bond stretch of the *a*-C/Si interface was obtained from

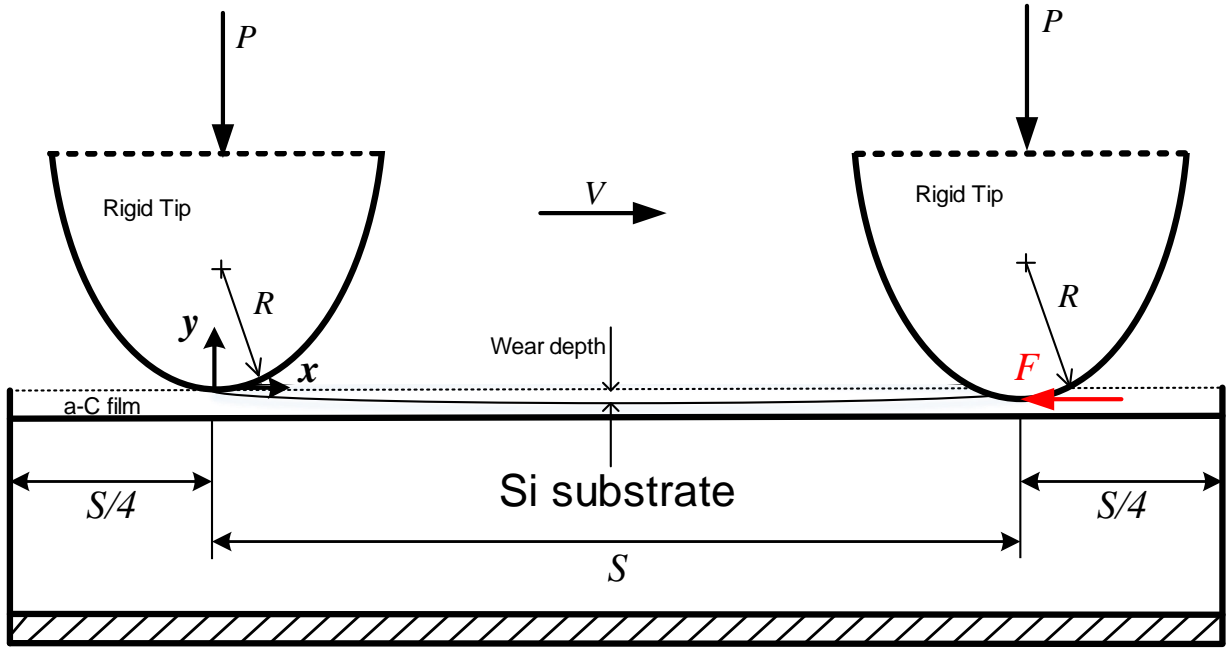


Figure 3.2: Schematic of peridynamics wear model of a sharp coniospherical diamond (rigid) tip under a normal load  $P$  sliding at constant velocity  $V$  and plowing through a thin  $a$ -C film, which is firmly attached to a thick Si substrate. The tip slides against the film surface from left ( $x/S = 0$ ) to right ( $x/S = 1$ ) by a total distance  $S$ . The shaded layer at the bottom of the substrate is modeled as rigid. The film thickness and the tip radius are not drawn to scale.

Eq. (2.42), where  $E$  is the elastic modulus of the substrate and  $G_I$  is the strain energy release rate due to film indentation by the coniospherical diamond indenter, which is equal to  $0.037 \text{ J/m}^2$  (Volinsky *et al.*, 2002; Marshall & Evans, 1984). Using Eq. (2.42) the critical stretch of the  $a$ -C/diamond interface was found to be equal to 0.007. The depth of the wear track on the film surface was determined by calculating the average displacement of irreversibly deformed ( $0 < D < 1$ ) particle layers of the film medium along the plowing path after unloading of the probe tip.

### 3.3 Results and discussion

Simulation results obtained with the peridynamics friction and wear models are presented in this section in conjunction with experimental results from a previous experimental study (Lu & Komvopoulos, 2001) to validate both peridynamics models.

Table 3.4: Coefficient of friction results from  $m$ -convergence tests.

| $\delta$ (nm) | $m$ | Coefficient of friction |
|---------------|-----|-------------------------|
| 8             | 3.5 | 0.1421                  |
| 8             | 4.0 | 0.1526                  |
| 8             | 4.5 | 0.1567                  |
| 8             | 5.0 | 0.1591                  |
| 8             | 5.5 | 0.1456                  |

### 3.3.1 Coefficient of friction

The coefficient of friction is defined as the ratio of the tangential (friction) force and the applied normal load. At each time step, the friction force was calculated as the tangential component of the total force generated by the sliding action of the tip; thus, a coefficient of friction was computed at each time step. An overall coefficient of friction was calculated for each  $a$ -C film as the average of all friction coefficient data.

Similar to local numerical methods, determining an appropriate grid size in peridynamics requires convergence testing (Bobaru *et al.*, 2009; Bobaru & Hu, 2012). Among various convergence tests, the  $m$ -convergence test was used in the peridynamics friction model to calculate the coefficient of friction. The  $\delta$ -convergence test was not used because the decrease of the horizon radius to zero (i.e., no length scale) should yield solutions converging to classical elasticity solutions, which not only do not hold at the nanoscale (Luan & Robbins, 2005) but are also length-scale independent. In the  $m$ -convergence test,  $\delta$  is fixed whereas  $m = \delta/\Delta x$  is gradually increased until the solution converges to an exact nonlocal peridynamics solution obtained for fixed  $\delta$  (Bobaru *et al.*, 2009). Table 3.4 shows results from  $m$ -convergence tests for film #7 (Table 6.1),  $P = 400 \mu\text{N}$ ,  $\delta = 8 \text{ nm}$ , and  $m$  in the range of 3.5-5.5. The coefficient of friction diverges with the increase of  $m$  from 5.0 to 5.5, suggesting an increasing effect of long-range forces. Therefore,  $m = 5.0$  (i.e.,  $\Delta x = 1.6 \text{ nm}$ ) was used in the present peridynamics analysis. Because the focus of this study is the analysis of thin  $a$ -C films, the convergence test was only carried out for the  $a$ -C film.

Table 3.5 shows a comparison between peridynamics and experimental results of the coefficient of friction of  $a$ -C films with different thickness and rms roughness for  $P$  in the range of 50-400  $\mu\text{N}$ . Even though the films were modeled to have ideally smooth surfaces, the agreement between numerical and experimental results is very good. It is noted that the experimental coefficients of friction represent averages of 300 data acquired along the entire sliding track and that the scatter in the measurements increases with the decrease of the normal load, yielding standard deviation values in the range of 0.05–0.08 (Lu & Komvopoulos, 2001).

Fig. 3.3 shows a comparison between peridynamics and experimental results of a typical



Table 3.5: Peridynamics and experimental results of the coefficient of friction of *a*-C films versus film thickness, roughness, and normal load.

| Film No. | Thickness <sup>(a)</sup> (nm) | Roughness <sup>(a)</sup> , rms (nm) | Normal load <sup>(a)</sup> ( $\mu$ N) | Coefficient of friction |                             |
|----------|-------------------------------|-------------------------------------|---------------------------------------|-------------------------|-----------------------------|
|          |                               |                                     |                                       | Peridynamics            | Experimental <sup>(a)</sup> |
| 1        | 31                            | 0.51                                | 50                                    | 0.132                   | 0.16                        |
|          |                               |                                     | 100                                   | 0.127                   | 0.14                        |
|          |                               |                                     | 200                                   | 0.115                   | 0.13                        |
|          |                               |                                     | 400                                   | 0.108                   | 0.12                        |
| 2        | 34                            | 0.20                                | 50                                    | 0.149                   | 0.17                        |
|          |                               |                                     | 100                                   | 0.128                   | 0.15                        |
|          |                               |                                     | 200                                   | 0.110                   | 0.14                        |
|          |                               |                                     | 400                                   | 0.099                   | 0.12                        |
| 3        | 39                            | 0.15                                | 50                                    | 0.156                   | 0.18                        |
|          |                               |                                     | 100                                   | 0.142                   | 0.16                        |
|          |                               |                                     | 200                                   | 0.121                   | 0.14                        |
|          |                               |                                     | 400                                   | 0.115                   | 0.13                        |
| 4        | 53                            | 0.27                                | 50                                    | 0.145                   | 0.17                        |
|          |                               |                                     | 100                                   | 0.127                   | 0.15                        |
|          |                               |                                     | 200                                   | 0.118                   | 0.13                        |
|          |                               |                                     | 400                                   | 0.102                   | 0.12                        |
| 5        | 59                            | 0.23                                | 50                                    | 0.140                   | 0.17                        |
|          |                               |                                     | 100                                   | 0.122                   | 0.15                        |
|          |                               |                                     | 200                                   | 0.114                   | 0.14                        |
|          |                               |                                     | 400                                   | 0.105                   | 0.12                        |
| 6        | 69                            | 0.15                                | 50                                    | 0.169                   | 0.18                        |
|          |                               |                                     | 100                                   | 0.148                   | 0.16                        |
|          |                               |                                     | 200                                   | 0.129                   | 0.14                        |
|          |                               |                                     | 400                                   | 0.121                   | 0.13                        |
| 7        | 95                            | 0.24                                | 50                                    | 0.147                   | 0.17                        |
|          |                               |                                     | 100                                   | 0.108                   | 0.15                        |
|          |                               |                                     | 200                                   | 0.115                   | 0.13                        |
|          |                               |                                     | 400                                   | 0.108                   | 0.12                        |

<sup>(a)</sup>Ref. (Lu & Komvopoulos, 2001)

coefficient of friction response for a 400  $\mu$ N normal load. The peridynamics solution closely follows the experimental trend, showing good agreement with the average response of the scattered experimental data. Fig. 3.3 and Table 3.5 illustrate the validity of the peridynamics friction model and provide justification for the modeling assumptions.

Fig. 3.4 shows peridynamics results of the steady-state coefficient of friction (obtained as the average of all numerical data in the 0-4  $\mu$ m sliding distance range) versus normal load for different *a*-C films (Table 6.1). All peridynamics solutions show that the coefficient of friction decreases with increasing normal load. This trend is in good agreement with

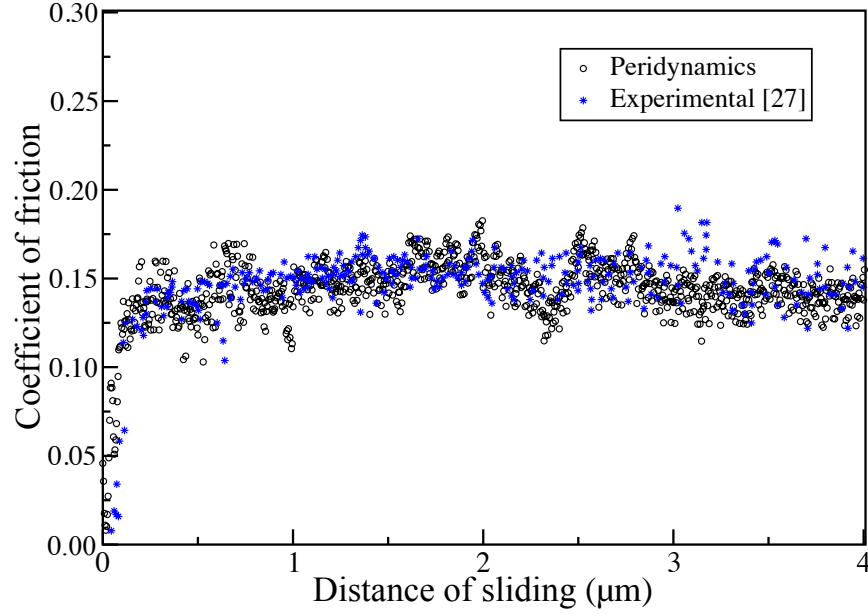


Figure 3.3: Peridynamic and experimental results of the coefficient of friction of an *a*-C film versus sliding distance for  $P = 400 \mu\text{N}$ .

experimental findings and is attributed to the inverse proportionality of the coefficient of friction of predominantly elastically deformed surfaces to the cubic root of the normal load (Lu & Komvopoulos, 2001).

Figs. 3.5a and 3.5b, respectively, show  $y$ - and  $x$ -displacement contour maps for  $x/S = 1$ . High-magnification views of the particle displacements under the tip, shown in Figs. 3.5c and 3.5d, provide insight into the highly stressed region of the film underneath the loaded tip. However, the zero displacements in the wake of the tip (Figs. 5a and 5b) reveal the full recovery of the unloaded film region. Importantly, despite the fully elastic behavior of the film/substrate medium and the adhesionless tip/film contact interface, the instantaneous coefficient of friction is not zero. Not only is this finding in contrast with classical friction theories, which attribute friction to irreversible deformation, but also reveals that frictionless contact is practically impossible.

### 3.3.2 Wear depth

To validate the peridynamics wear model, the experimentally measured depths of wear tracks produced on *a*-C film surfaces by a sharp conospherical rigid tip (Lu & Komvopoulos, 2001) are compared with numerical results. Fig. 3.6a shows damage at the film surface due to sliding contact interaction. A damage index  $D = 1$  is indicative of bond breakage. As the rigid tip plows through the film, the particles near the surface are permanently displaced from their original positions. Fig. 3.6b shows a close-up view of the deformed grid below the tip, revealing much more pronounced damage in the near-surface region of the *a*-C film adjacent

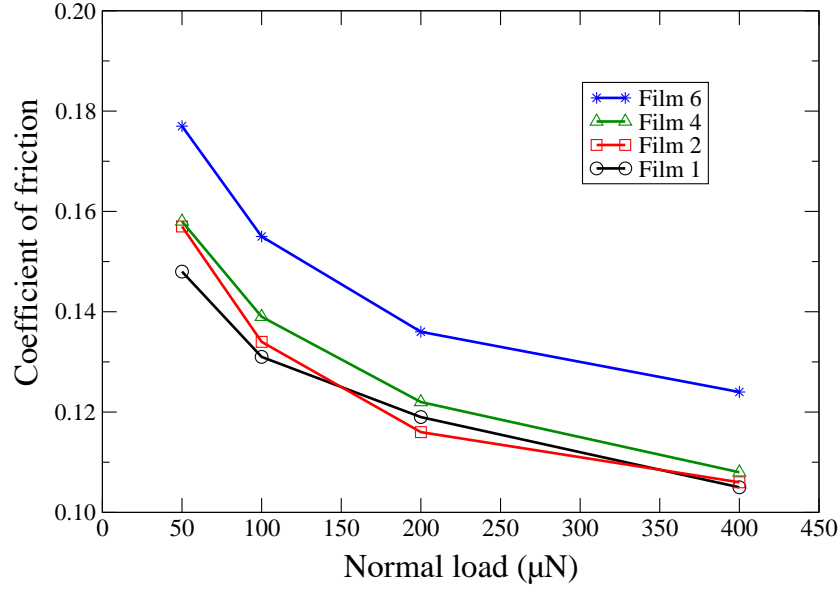


Figure 3.4: Peridynamic results of the coefficient of friction of various  $a$ -C films versus normal load.

to the sharp tip. Further insight into nanoscale film wear is provided by the  $y$ -displacement contour map shown in Fig. 3.6c and the close-up view of the near-surface damaged particle layers shown in Fig. 3.6d.

The wear depth is defined as the average displacement of all particles with partly broken bonds (i.e.,  $0 < D < 1$ ), located relatively far from the initial ( $x/S = 0$ ) and final ( $x/S = 1$ ) tip positions. The displacements of particles with  $D = 1$  were not included in the calculation of the wear depth. Table 3.6 shows numerical and experimental results of the wear depth of three  $a$ -C films for  $P = 10 \mu N$ . For films #8 and #9 the agreement is very good ( $\sim 4\%$  error), whereas for film #10 the agreement is fair, presumably because of the rougher film surface and errors due to the small film thickness (10 nm) relative to the grid size ( $\approx 1.6$  nm).

Table 3.6: Comparison of peridynamics and experimental results of the wear depth of  $a$ -C films versus film thickness, roughness, and normal load

| Film No. | Thickness (nm) | Roughness, rms (nm) | Normal load ( $\mu N$ ) | Wear depth (nm) |                             |
|----------|----------------|---------------------|-------------------------|-----------------|-----------------------------|
|          |                |                     |                         | Peridynamics    | Experimental <sup>(a)</sup> |
| 8        | 17             | 0.19                | 10                      | 0.730           | 0.76                        |
| 9        | 22             | 0.18                | 10                      | 0.391           | 0.40                        |
| 10       | 10             | 0.20                | 10                      | 0.161           | 0.20                        |

<sup>(a)</sup>Ref. (Lu & Komvopoulos, 2001)

Nanoscale material behavior can be challenging because classical continuum theory breaks down at the nanoscale, whereas MD analysis is limited to very small scales of limited prac-

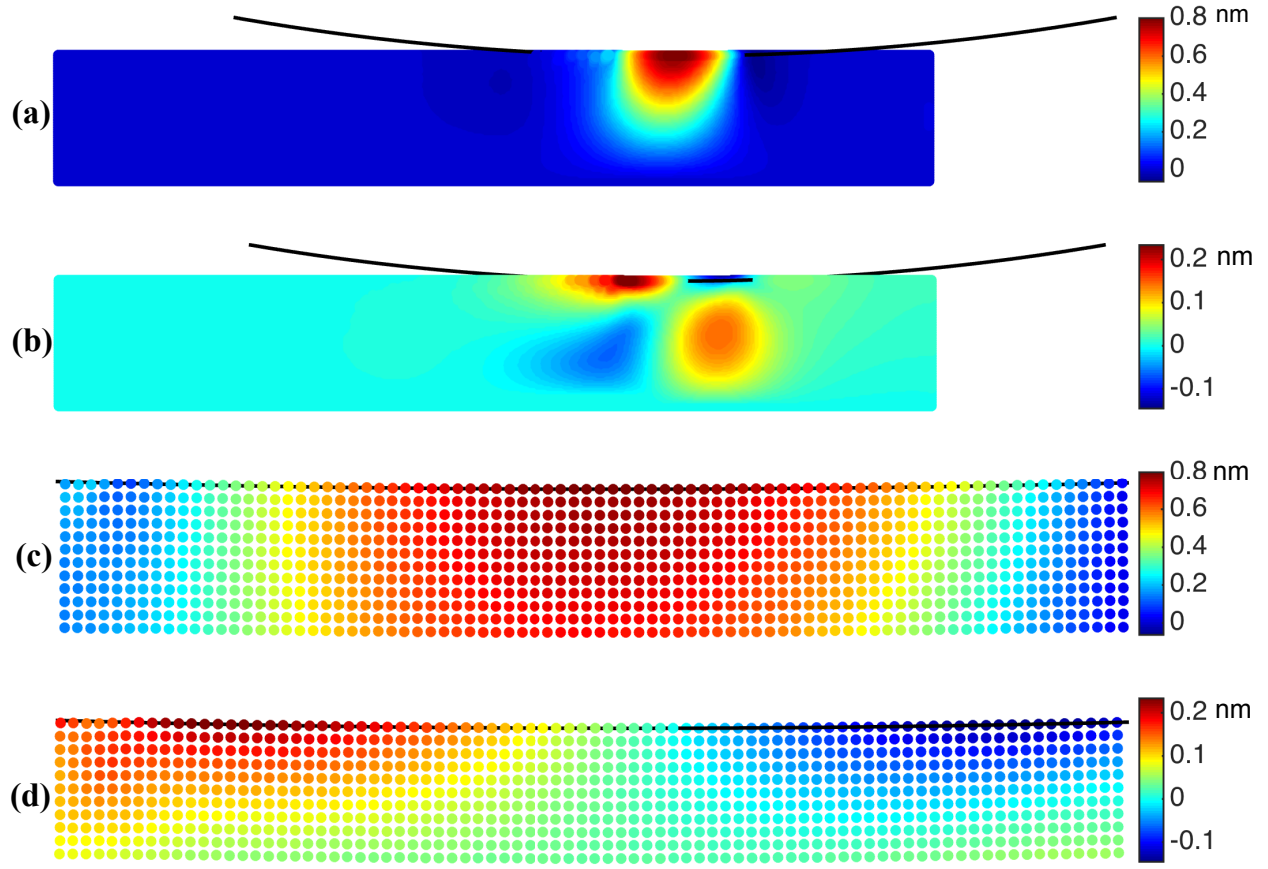


Figure 3.5: (a), (b) Contour maps of  $y$ - and  $x$ -displacements of particles in film #4 for  $x/S = 1$  and (c), (d) corresponding high-magnification contour maps showing the displacement of particles in the near-surface region of the film adjacent to the contact interface with the sliding rigid tip.

tical use. Peridynamics promises to bridge the material gap in computational mechanics. In this study, state-based peridynamics theory was used to develop friction and wear models of thin films. Favorable comparisons between peridynamics solutions and experimental results of thin films obtained under identical testing conditions illustrate the validity of the peridynamics friction and wear models developed in this study. Long-range forces were not considered and film surfaces were modeled as perfectly smooth, because the film roughness is significantly less than the grid size. Despite these assumptions, very good agreement was obtained between peridynamics and experimental results of films with thickness equal or larger than 10 nm. The results of this study demonstrate the potential of peridynamics to capture the nanoscale tribological behavior of thin films, which is difficult (if not impossible) to achieve at the nanoscale with other numerical techniques, such as finite element and boundary elements methods.

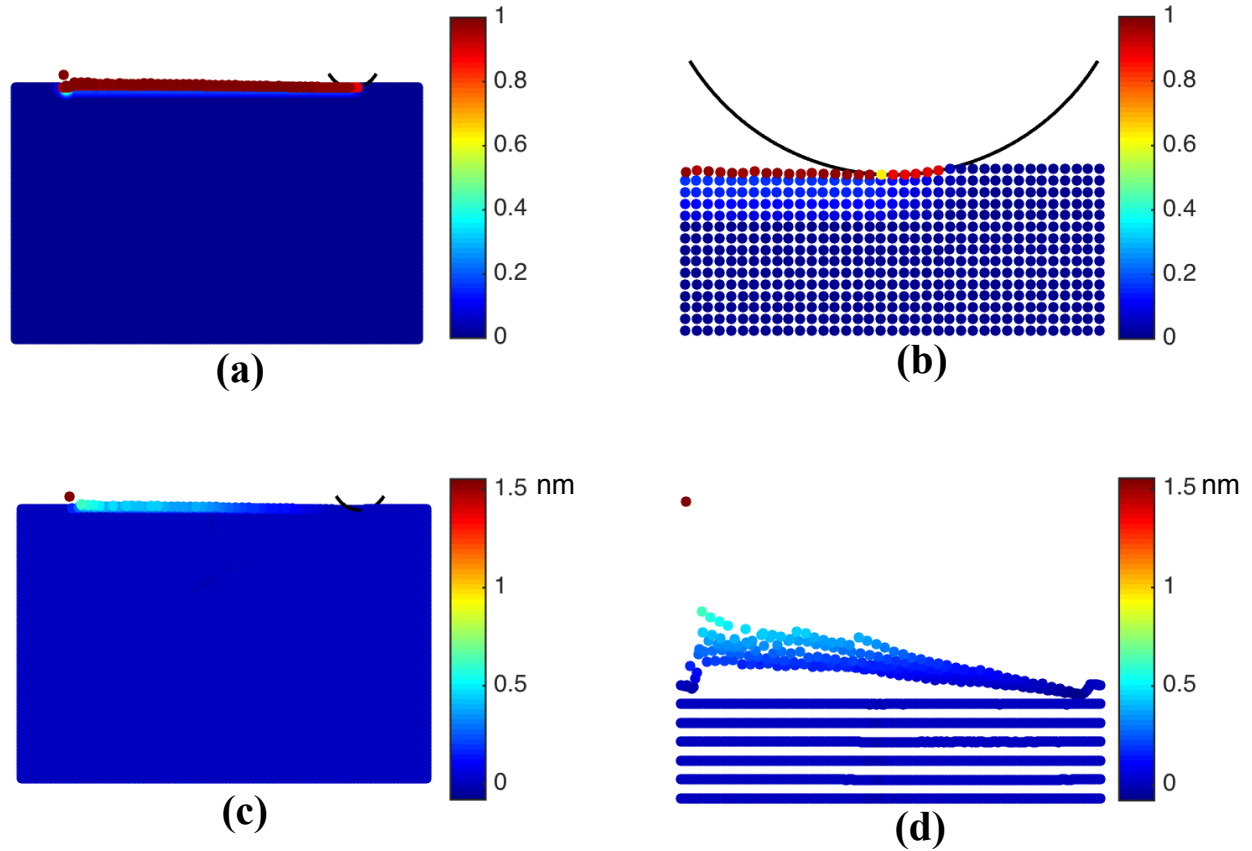


Figure 3.6: (a) Damage contour map of film #9 after tip unloading (damage index  $D = 1$  corresponds to particles with all their bonds broken), (b) corresponding high-magnification damage contour map of damaged particle layers in the near-surface region of the film, (c)  $y$ -displacement contour map of damaged film #9 after tip unloading, and (d) corresponding high-magnification contour map showing the  $y$ -displacement of particles adjacent to the contact interface with the plowing rigid tip.

## Chapter 4

# Peridynamic Modeling of Delamination in Thin Compressed Films

### 4.1 Introduction to Delamination

Laminated composites are being utilized as the preferred material various industries, such as marine, aerospace, civil, and automotive structures, micro-electromechanical systems (MEMS). Their structural integrities such as high toughness, strength, corrosion-resistance, and stiffness for a given loading scenario, have been the dominant reasons for them to replace their major conventional metallic counterpart. Their major role in the industrial application highlights the need for studying their behavior and failure mechanism to continue and advance their practicality and use in high demanding technologies.

The premier failure mechanism in layered composites, specially the thin film structures, is the separation of the layers that occurs along their interface. This phenomenon is called *delamination* and has been widely studied since early 40s when it was first observed in the buckling of sandwich panels in aircraft structures (Gough *et al.*, 1940). More recently, failure modes of films/substrate systems have been investigated for various applications such as wear resistant coatings (Burnett & Rickerby, 1987), ceramic thermal barrier coatings in aircraft and automobile industries (GEIGER, 1992), stretchable electronics (Khang *et al.*, 2006), hard transparent coatings on optical polymers (Gioia & Ortiz, 1997), etc.

Delamination is known to occur as a result of increased interlaminar tension and shear that give rise to the development of discontinuities such as wrinkling, buckling, channel cracks or free edge effects. This condition might be met either under working or manufacturing conditions. Therefore, it is very important to inspect such materials with excessive care in various loading and environmental conditions to detect any abnormality that might remain invisible in a particular setting while leading to catastrophic failures once they are deployed in a certain application. Delamination can significantly decrease the mechanical integrities until it eventually causes complete failure either alone or together with other failure mechanisms.

Studying and analyzing the thin-layered composites has been performed both numerically



Figure 4.1: Deformation mechanisms in compressed thin films. From left to right: wrinkling with no delamination, buckling-driven interface delamination, concomitant wrinkling and buckle delamination

and experimentally. Experimental methods despite of resulting realistic solutions, are often limited to precise controlled conditions, time consuming, and expensive where numerical methods do not have these limitations if are built upon realistic assumptions. In either direction, it is essential to understand the governing principles of initiation, growth, and mixture of delamination with other failure modes.

#### 4.1.1 Deformation mechanisms in compressed thin films

In this work we focus only on isotropic elastic thin film composite structures under lateral compression because they are arguably the most dominantly used thin film structures. Due to the significant mismatch in the material properties of these thin film and the compliant substrate, different deformation mechanisms can occur at the film/substrate interface such as wrinkling, wrinkle-induced delamination, buckling, and concomitant wrinkling and buckling (in 2D), and channel cracking of the thin film (in 3D) which are the most commonly observed ones. Figure 4.1 shows the three dominant delamination modes in 2D thin film structures in compression. It is important to note that these mechanisms are not always considered as failure modes. Researchers have been able to manipulate them to obtain the desired patterns for creative applications ([Harrison \*et al.\*, 2004](#); [Chan \*et al.\*, 2008](#)).

#### 4.1.2 Why Peridynamic?

In conventional continuum mechanics, interfacial delamination modes have been treated with either linear elastic fracture mechanics (LEFM) and cohesive zone modeling (CZM). The biggest downside of the LEFM method is that it cannot predict crack initiation. In other words, LEFM assumes a pre-defect exists and it grows from there. Moreover, LEFM assumes only small-scale yielding around the crack tip and elastic behavior for the rest of the body. Furthermore, micromechanisms of fracture are ignored in LEFM. Comprehensive reviews on the fracture mechanics of different modes of interfacial cracks have been provided by Rice ([Rice, 1988](#)) and Hutchinson ([Hutchinson & Suo, 1991](#)).

Because the aforementioned assumptions of LEFM do not always remain valid, a non-linear elastic approach, CZM in this case, has been also used to predict the delamination modes of film/substrate systems. CZM method considers the damage at microscopic level by introducing the bridging law in the bridging zone ahead of the crack tip that can be derived analytically or experimentally for different types of elastic or elastic-plastic materials. However, similar to the small-scale plastic zone in LEFM, the bridging zone in CZM is estimated under the condition of small-scale bridging that has to remain small compared to the crack size (Mei, 2011). Similar to other approaches of conventional continuum mechanics framework, solving partial differential equations (PDEs) is necessary in CZM which enforces mathematical difficulties in the presence of defects as they become singular. Most importantly, in order to study concomitant wrinkling and buckling or crack channels in conventional theory, one must assume that a pre-crack already exists at the interlayer interface whereas in many applications this assumption cannot be afforded. Moreover, wrinkling mechanisms is essentially a crack-free mechanism that is impossible to be modeled realistically with conventional theory. In PD theory, failure is part of the solution not the input to the problem. Therefore, a material fails where it means to fail, not where we introduce weakness to it. Furthermore, PD employs integral of differential equations instead of PDEs and thus no mathematical complexity is enforced. Hence, cracks and all other types of defects can be treated as any other part of the material.

This chapter is partially inspired by a previously published thesis (Mei, 2011) which had a review on analytical solution to the fracture and delamination of elastic thin films and using finite element modeling. We are particularly interested in answering following question in modeling delamination in thin films using PD:

- Under what circumstances deformation mechanisms change in compressed thin films?
- What are the controlling factors in surface patterns generated on thin films?
- How to quantitatively predict evolution of surface patterns and/or failure modes?

## 4.2 Modeling deformation mechanisms in thin films

Our goal is to develop a model that can predict deformation patterns on the thin film given the mismatch between the mechanical properties of the film and the substrate. Figure 4.2 shows these properties as the controlling factors for surface patterns generated on the thin film. We want to explore how materials' properties mismatch can translate into change in the materials' behavior as the structure undergoes various compressive strain.

### 4.2.1 Wrinkling without delamination

We first consider the case where the elastic thin film is placed on an elastic thick substrate, as shown in the leftmost pattern in Fig. 4.1. The structure undergoes lateral compressive



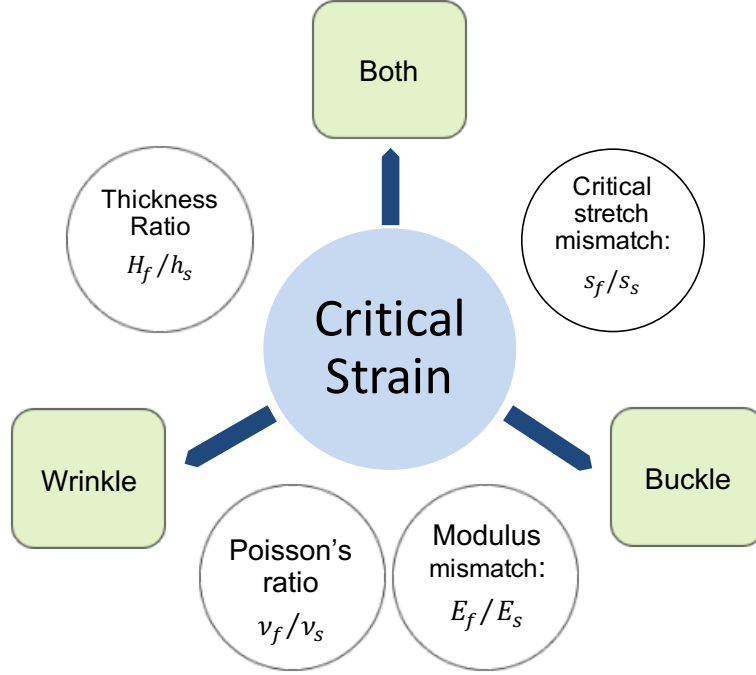


Figure 4.2: Controlling parameters for surface patterns on the thin compressed film on a compliant substrate

strain which results in wrinkling behavior of the film without being separated from the compliant substrate. We begin with providing the analytical solution followed by our PD analysis of this problem and their comparison.

#### 4.2.1.1 Analytical solution

A widely-known analytical solution for the critical strain for wrinkling to initiate is given below which was introduced in (Chen & Hutchinson, 2004; Mei, 2011). It is derived based on assuming that the shear traction at the interface is zero.

$$\epsilon_{cr}(kh) = \frac{(kh)^2}{12} + \frac{1}{2kh} \frac{\bar{E}_s}{\bar{E}_f} \quad (4.1)$$

where subscripts  $f$  and  $s$  represent film and substrate properties, respectively,  $\bar{E} = E/(1 - \nu^2)$  denotes the plane-strain modulus with  $E$  as the Young's modulus and  $\nu$  as the Poisson's ratio.  $k$  is the wave number such that the wrinkle wavelength is  $\lambda = 2\pi/k$ . At  $\lambda^* = 2\pi h \left( \frac{\bar{E}_f}{3\bar{E}_s} \right)^{1/3}$ , the minimum required critical strain to onset wrinkling is

$$\epsilon_{cr}^* = \frac{1}{4} \left( \frac{3\bar{E}_s}{\bar{E}_f} \right)^{2/3} \quad (4.2)$$

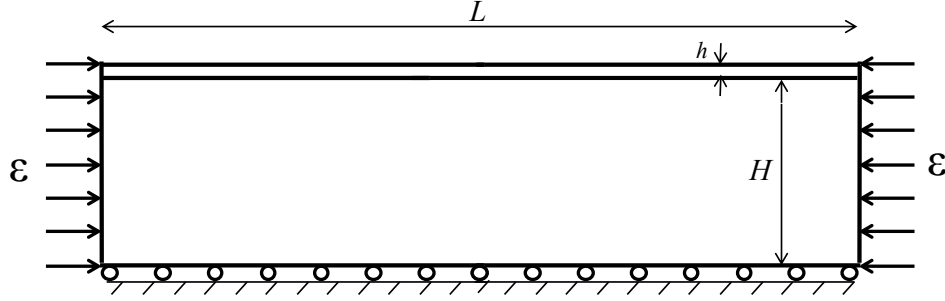


Figure 4.3: Schematic of the thin film and the compliant substrate model

For a compressive substrate ( $\nu_s < 0.5$ ), the shear traction can no longer be assumed as zero and hence Eq. (4.2) is violated. However, most of the engineering materials have Poisson's ratio values of less than half. (Mei, 2011) introduced a more accurate analytical solution using linear plate equations for this problem in which the effect of shear traction is taken into account. In this solution, the tractions acting on the substrate surface are assumed to be periodic tangential  $\tau = \tau_m \sin(kx)$  and normal  $q = q_m \cos(kx)$ . By solving the linear elasticity equations under the plane-strain condition together with equilibrium condition of the film the critical strain was found as below:

$$\epsilon_{cr}(kh) = \frac{(kh)^2}{12} + \frac{1}{2kh} \left( \frac{\bar{E}_s}{\bar{E}_f} \right) \left[ 1 - \frac{1}{2} \left( \frac{1 - 2\nu_s}{1 - \nu_s} \right)^2 \left( \frac{\bar{E}_s}{\bar{E}_f} \frac{1}{kh} + 2 \right)^{-1} \right]^{-1} \quad (4.3)$$

This critical strain is minimized at wavelength of  $\lambda^*$  given below:

$$\lambda^* = 2\pi h \left( \frac{\bar{E}_f}{3\bar{E}_s} \right)^{1/3} \left[ 1 - \frac{1}{4} \left( \frac{1 - 2\nu_s}{1 - \nu_s} \right)^2 \right]^{1/3} \quad (4.4)$$

which results in the minimum critical strain of

$$\epsilon_{cr}^* = \frac{1}{4} \left( \frac{3\bar{E}_s}{\bar{E}_f} \right)^{2/3} \left[ 1 - \frac{1}{4} \left( \frac{1 - 2\nu_s}{1 - \nu_s} \right)^2 \right]^{2/3} \quad (4.5)$$

#### 4.2.1.2 Peridynamic delamination model

A schematic of elastic thin film and compliant substrate is illustrated in Fig. 4.3. They are both assumed to be elastic and isotropic and undergo small lateral compression in a plane-strain condition.  $L$  is length and  $h$  and  $H$  correspond to the film and substrate thicknesses. Our goal is to determine the minimum strain required to initiate wrinkling in the thin film assuming that there is no interfacial pre-defect of any kind present. The film and substrate are both modeled uniformly with 2D quadrilateral elements. In our peridynamic model, lateral displacement ( $\bar{u}$ ) is applied on both layers through the *fictitious layer* (Madenci & Oterkus, 2014). Fictitious layer is composed of imaginary particles which are excluded from

Table 4.1: Material properties of the film and substrate in analysis of deformation mechanisms

| Material properties          | Film  | Substrate |
|------------------------------|-------|-----------|
| Young's Modulus (GPa)        | 200   | 2         |
| Poisson's ratio              | 0.33  | 0.33      |
| Density (kg/m <sup>3</sup> ) | 8000  | 8000      |
| Thickness                    | 96 mm | 4mm       |

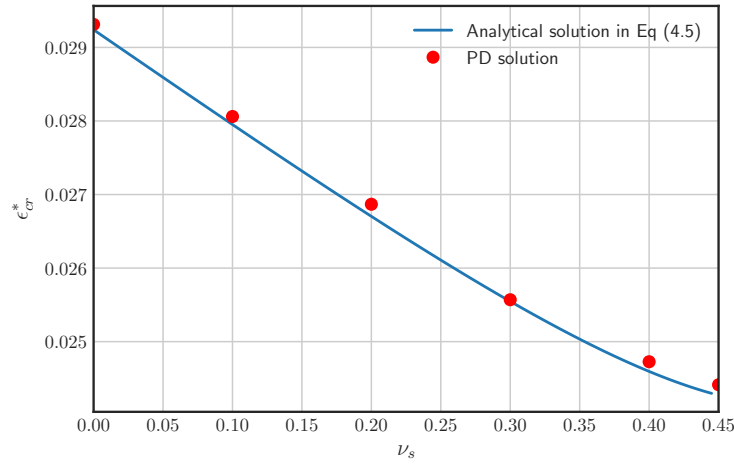


Figure 4.4: Comparison between analytical solution and PD for the critical strain required to onset wrinkling without delamination for various Poisson's ratios of the substrate.

PD force calculation but exist to transmit the boundary conditions to the real particles in a realistic way such that immediate or excessive damage does not occur at boundaries where external displacement, velocity or force is prescribed. Here a layer with a width of horizon ( $3.015\Delta x$ ) is reserved from each side. The applied displacement on the left and right boundaries are normalized by  $L$  to attain the required strain ( $\epsilon = \bar{u}/L$ ) as the solution to the problem.

Figure 4.4 shows the comparison between the analytical solution presented in Eq. (4.5) and PD solution found for the critical strain to onset wrinkling without causing delamination. Both methods follow the same path and intuitively agree that as the substrate becomes more compressive, it is easier to shape wrinkles with it.

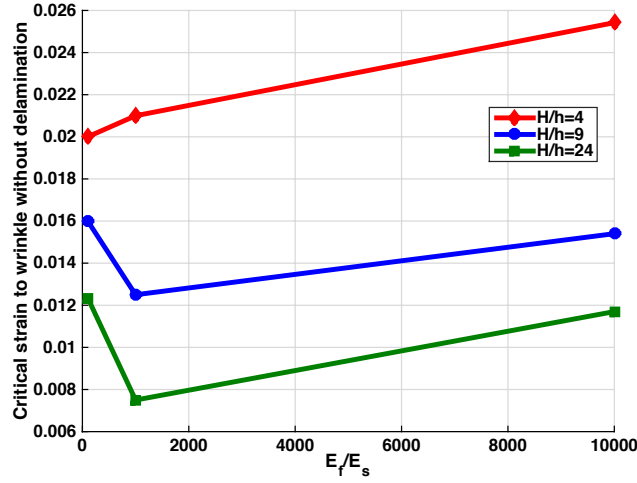


Figure 4.5: Effect of elastic moduli mismatch on onset of wrinkling with no delamination for different film and substrate thicknesses ratios

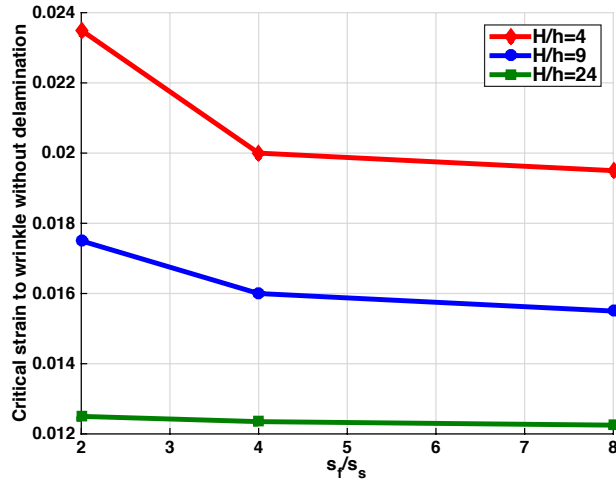


Figure 4.6: Effect of critical bond stretch mismatch on onset of wrinkling with no delamination for different film and substrate thicknesses ratios

### 4.2.2 Effect of mechanical properties mismatch

After validating our PD implementation with analytical solution, we can now analyze the playing factors in surface pattern generation of the film in this scenario. Figure 4.5 illustrates the effect of the elastic moduli mismatch. It is interesting to note that for large thicknesses ratio ( $H/h$ ), we see a drop in required strain to onset wrinkling for Elastic moduli mismatch (around 1000) whereas for smaller values of  $H/h$ , the required strain increases as the substrate becomes more compliant.

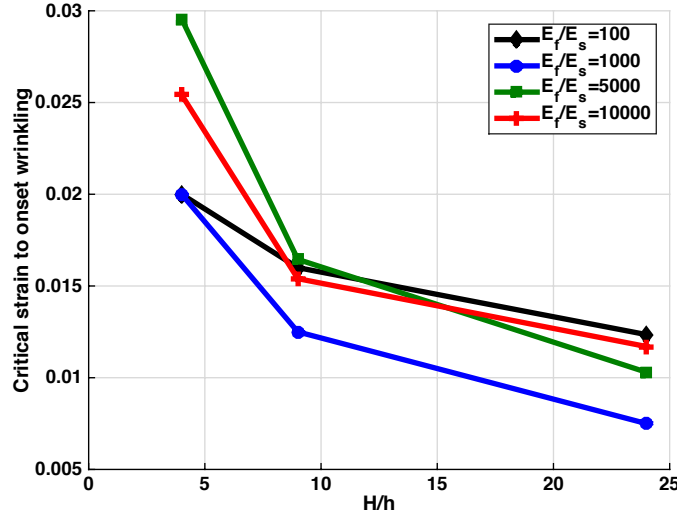


Figure 4.7: Effect of layers' thickness mismatch on onset of wrinkling with no delamination for different elastic moduli mismatches

### 4.2.3 Wrinkling with delamination

As it was found in the previous scenario, the required strain to onset wrinkling is very low and beyond it we observe different behaviors in the film. In order to track this behavior using conventional theory, one might calculate the wrinkling amplitude. As it grows and reaches a critical value, it is expected to see the normal and shear tractions acting on the film/substrate interface increase, which may result in delamination (Mei *et al.*, 2011; Shield *et al.*, 1994; Goyal *et al.*, 2010). However this interfacial tractions are assumed to occur in nonlinear post-buckling regime where zero tangential displacement is assumed so that the maximum shear traction is linearly related to the maximum normal traction. (Mei *et al.*, 2011) derives the maximum normal traction as a function of nominal strain as follows:

$$q_m = \frac{4(1 - \nu_s)^2 \bar{E}_s}{3 - 4\nu_s} \sqrt{\epsilon - \epsilon_{cr}} \quad (4.6)$$

In order to compute the critical strain required to onset wrinkling with delamination, (Mei *et al.*, 2011) followed the solution offered by a cohesive zone model by (Hutchinson & Evans, 2000) in which the the maximum normal traction in Eq. (4.6) is set equal to the peak stress in the normal traction-separation relation for the interface ( $\sigma_{int}$ ). The critical strain to induce wrinkling with delamination is found as

$$\epsilon_{cr}^{**} = \epsilon_{cr}^* + \left( \frac{3 - 4\nu}{4(1 - \nu_s)^2} \frac{\sigma_{int}}{\bar{E}_s} \right)^2 \quad (4.7)$$

where  $\epsilon_{cr}^*$  is given in Eq. (4.5). Beyond  $\epsilon_{cr}^{**}$  it is expected to observe delamination and wrinkling to co-exists and interact with each other.

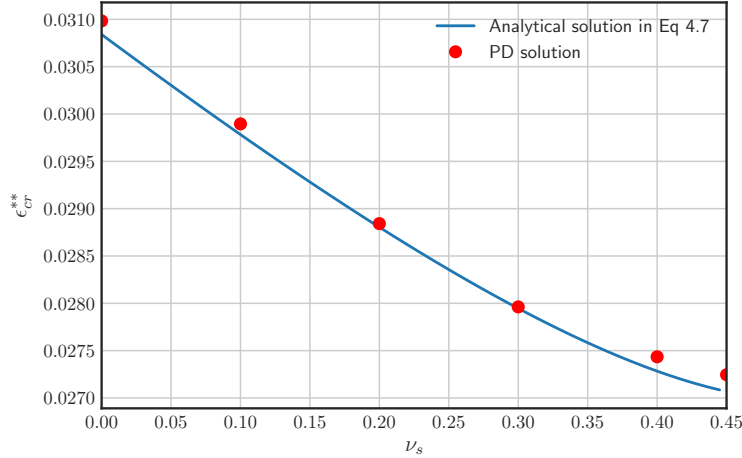


Figure 4.8: Comparison between analytical solution and PD for the critical strain required to onset wrinkling with delamination for various Poisson's ratios of the substrate.

It was also shown in (Mei *et al.*, 2011) that their FEA results for maximum normal traction closely agreed with Eq. (4.6). Therefore, it fair to only compare our PD results with the analytical solution in this scenario as well. ( $\sigma_{int}$ ) in PD solution is computed as the maximum PD force at the interface divided by the area occupied by a PD particle ( $\Delta x^2$ ).

Figure 4.8 shows a tight agreement between the PD and the analytical solutions.

### 4.3 Surface pattern prediction

Similar to the previous scenario, we can now generate surface patterns prediction graphs by plotting the critical strain versus mechanical properties mismatch. One of the most advantage of PD models is that they do not require any pre-assumption to be able to predict the film surface patterns. Different behaviors happen when the material is ready for them. This feature is illustrated in Figs. 4.9 and 4.10 where the transition between different behaviors is very clear. These graph can help predicting when different patterns are expected to be seen given material properties and the applied compressive strain.

Recall that we equated strain energy density concept (Eq. (2.21) in both classical and PD theories to derive the material constants in the PD force density function derivations. By looking into the strain energy function again for the film/substrate system as it goes under compression, we observed how changes in the strain energy are related to the changes in generated patterns on the film.

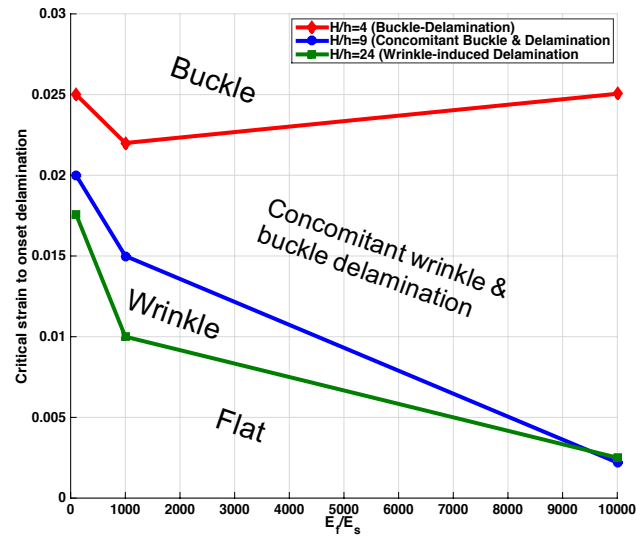


Figure 4.9: Effect of elastic moduli mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios

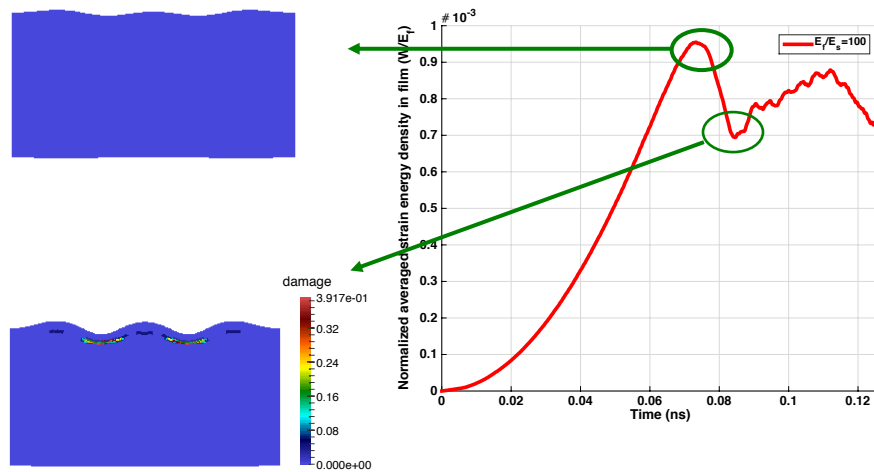


Figure 4.11: Energy density function for film/substrate undergoing compressive strain until wrinkling pattern generation

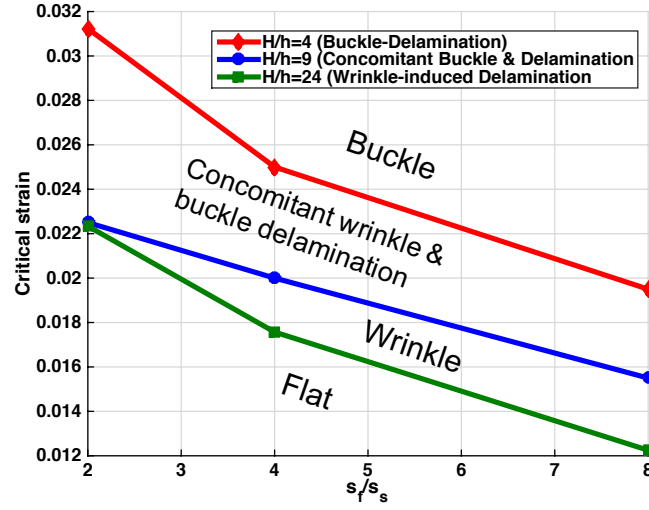


Figure 4.10: Effect of critical bond stretch mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios

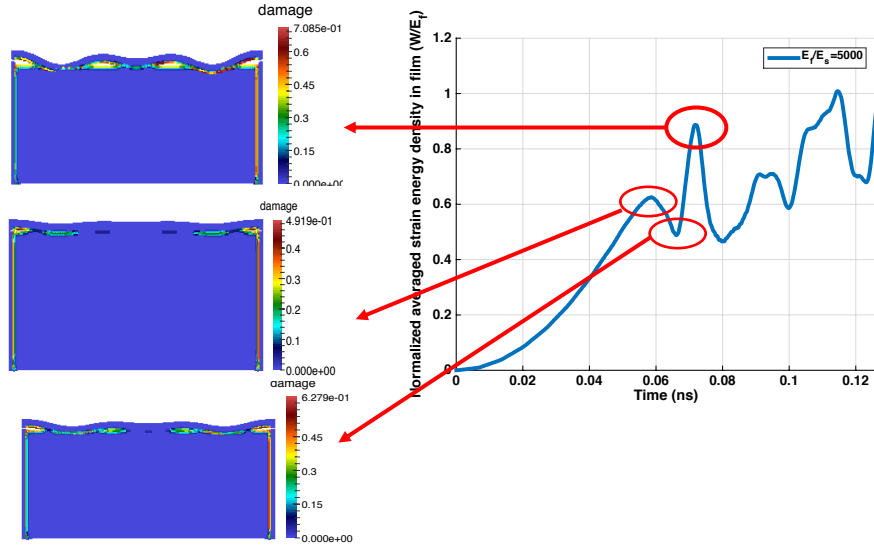


Figure 4.12: Energy density function for film/substrate undergoing compressive strain exhibiting concomitant wrinkling and buckling behavior

Figure 4.11 depicts the strain energy density function for a film/substrate system under-



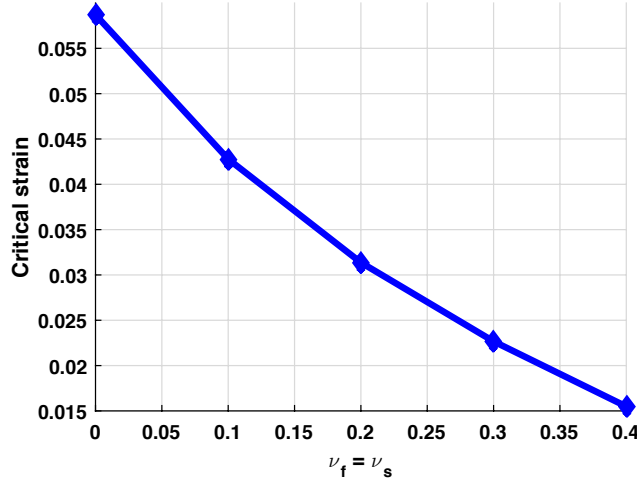


Figure 4.13: Effect of Poisson's ratio mismatch on onset of wrinkling with delamination for different film and substrate thicknesses ratios

going compressive strain until wrinkling. For significant changes in the energy function, one can see the film/substrate setup illustrated on the left corresponding to their strain energy function value. Intuitively the strain energy increases until the first signs of wrinkling start to appear. That is when the film is no longer able to resist deformation and the strain energy is maximized. As a response to the applied strain, the film will wrinkle and uses the stored energy density to provide energy for that. This explains the small drop in the energy function until it starts going back up again as more strain is introduced into the system. Sole wrinkle patterns are only present until this point and beyond this local minimum, wrinkles will be followed by delamination. The noisy behavior shown in the graph explains the nature of this phenomenon which is unstable because delamination is accompanied by breaking bonds at the interface.

In Fig. 4.12, we start with applying more initial strain to skip through wrinkling-only behavior and jumping into seeing damage at the interface. Similar to the above scenario, the first local maximum of this plot is also an indicative of change in the film pattern. Wrinkling with delamination appears first at this point and it continues jumping up and down depending on the damage happening at the interface. While it was interesting to see the correspondence between the first local maximum of this plot and the pattern change in the system, we have pointed out two other extreme points in the energy function and their film pattern correspondence.

## 4.4 Discussion

In the present study, we considered two scenarios for concomitant wrinkling and buckle-delamination of an elastic thin film on a compliant substrate under compression. First, if

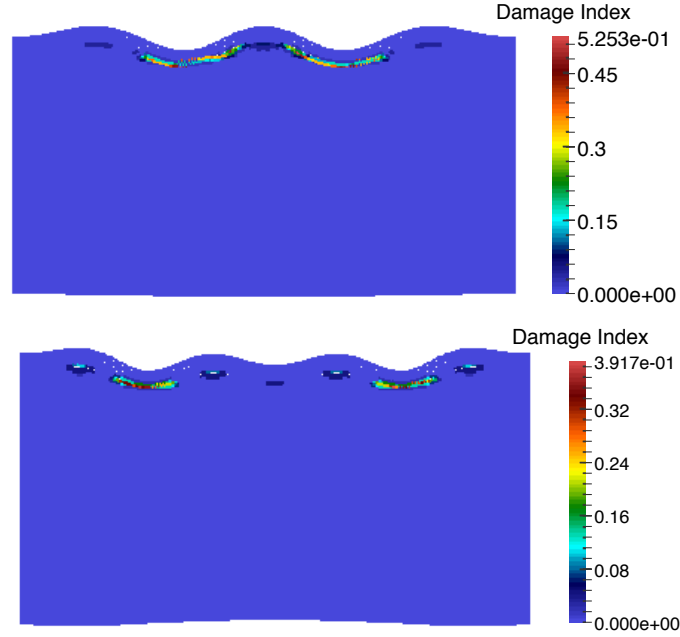


Figure 4.14

the film/substrate interface is perfectly bonded, wrinkling occurs beyond a critical strain. Subsequently, nucleation of interfacial delamination may occur at a larger nominal strain when the wrinkle-induced normal traction at the interface exceeds the strength of the interface. The growth of the interfacial delamination however requires further studies, for which a cohesive zone model may be adopted for the interface.

While studying the buckling phenomenon with classical theory requires knowledge about the pre-existing interfacial delamination, in PD theory, we do not need to deal with any pre-existing defect of any kind. These two scenarios qualitatively agree with the experimental observations (Mei *et al.*, 2011).

Two failure criteria are suggested for hybrid systems with stiff thin films on compliant substrates. First, we can compute the strain energy density of the particles at the interface. By taking the gradient of the strain energy function and find its roots, we will be able to determine the transition between the wrinkling, concomitant wrinkling and buckling, and buckling. The second criterion is to plot *phase diagrams* similar to Figs. 4.9 and 4.10 for critical strain versus mechanical properties mismatch. Every hybrid setup undergoing compressive strain, has a unique phase diagram that determines the change in mechanical behavior across different properties mismatch.

The present study utilizes peridynamic to model delamination and has overcome the aforementioned disadvantages of the conventional continuum mechanics and proposes PD as a more general theory in modeling delamination.

## Chapter 5

# Peridynamic for Plastic Deformation

Plastic deformation frequently occurs in engineering components when they are loaded beyond their elastic limit. Predicting plastic behavior is essential in design or in determining the cause of failure in presence or the absence of pre-existed flaws (voids, cracks, etc.).

Conventional continuum theory uses the assumptions of the elastic-plastic fracture mechanics (EPFM) to study plasticity in materials which undergo large deformations. EPFM considers the material as isotropic and elastic-plastic. Once the strain energy fields or opening displacement near the crack tips reach their critical limits, the crack path can be predicted. This is a well established theory and thoroughly discussed in ... However, it has some limitations that cannot be overcome with local theories.

Peridynamics (PD) is a nonlocal theory introduced by Silling ([Silling, 2000](#)) that uses the integral of differential equations of motion instead of partial differential equations which results in no mathematical difficulty associated with singularities. For elastic deformations, PD has been successfully applied in various areas such as fracture mechanics by ([Silling & Askari, 2005](#)), ([Agwai et al., 2008](#)), ([Ha & Bobaru, 2010](#)), friction and wear in thin films by ([Ebrahimi et al., 2015a](#)), analysis of impact damage by ([Xu et al., 2008](#)), and mechanical behavior of composite materials by ([Bobaru & Silling, 2004](#)), ([Askari et al., 2006](#)), ([Kilic et al., 2009](#)), and ([Hu et al., 2011](#)).

Peridynamics formulation was extended to be used for viscoplastic deformations by ([Foster et al., 2010](#)) where they incorporated rate-dependent plastic behavior in the non-ordinary state-based PD. Ordinary state-based plasticity framework was first proposed by ([Mitchell, 2011b](#)) where a flow rule for the perfect plasticity was proposed. Shortly after, ([Mitchell, 2011a](#)) added rate dependency to his previous work in the state-based formulation. ([Taylor, 2008](#)) used sub-layer method to account for the plastic and viscoplastic behavior of materials in the bond-based formulation. Recently, ([Madenci & Oterkus, 2016](#)) developed an ordinary state-based plasticity formulation based upon von Mises yield criterion with isotropic hardening. They also derived a peridynamic path-independent J-integral formulation.

This study presents a new approach in incorporating plasticity with ordinary state-based formulation of ([Madenci & Oterkus, 2014](#)) which uses mechanical sublayer method originally proposed by ([Besseling, 1953](#)) and extended by ([Zienkiewicz et al., 1973](#)) and ([Pian, 1987](#)).

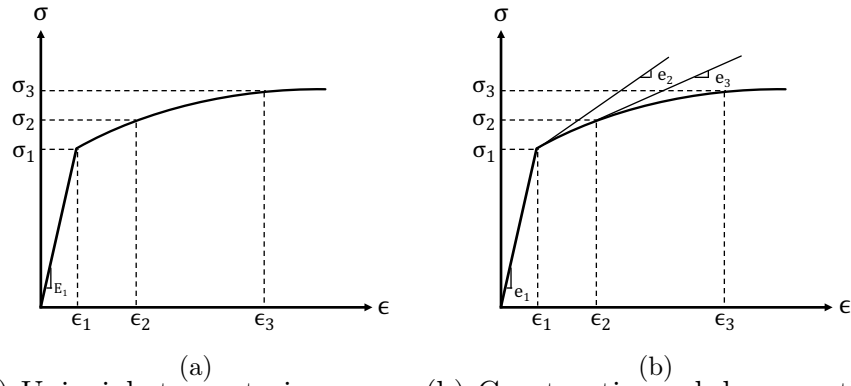


Figure 5.1: (a) Uniaxial stress-strain curve ; (b) Constructing sub-layer method with three layers

It differs from the previous work by (Taylor, 2008) in defining yield criterion and using more general form of state-based theory. Results obtained for J-integral calculation are compared with those from benchmark problems and an excellent agreement is achieved.

## 5.1 Plastic deformation

Mechanical sub-layer method introduced by (Besseling, 1953) and later extended by (Pian, 1987) and (Zienkiewicz *et al.*, 1973) models a time-independent elastic-plastic behavior with kinematic hardening. As the name suggests, it assumes the material is composed of  $n$  sub-layers, from which the first one is elastic whereas the following  $n - 1$  layers are elastic-plastic with different yield stresses. Given a stress-strain curve of the material, the method replaces it with a piecewise linear function and successively models the work hardening behavior of the material in different loading conditions. This suggests that having more number of sub-layers can lead to a more accurate plot of nearly linear segments.

The procedure of constructing the sub-layer model is as follows. Figure 5.1(a) shows a stress-strain curve for a typical ductile material. The curve is divided into  $n = 3$  sub-layers where the first segment ends at the initial yield stress ( $\sigma_1$ ) and the corresponding strain ( $\epsilon_1$ ). The slope of the first layer is clearly the Young's modulus. The following points ( $\sigma_2$ ) and ( $\sigma_3$ ) are also selected from this curve. In Fig. 5.1(b) the slopes of each segment is found to construct the sub-layer segments such that the resulting stress-strain curve intersects the original stress-strain curve at the previously chosen points. This figure clearly shows the more segments provide a more precise insight to the hardening behavior.

### 5.1.1 Peridynamics plasticity using sublayer method

We have adopted the idea of the sub-layer method to predict the plastic behavior of a ductile material once it experiences loadings above its elastic limit. In this method, the key idea is that a peridynamic bond is assumed to be composed of  $n$  sub-layers which are in

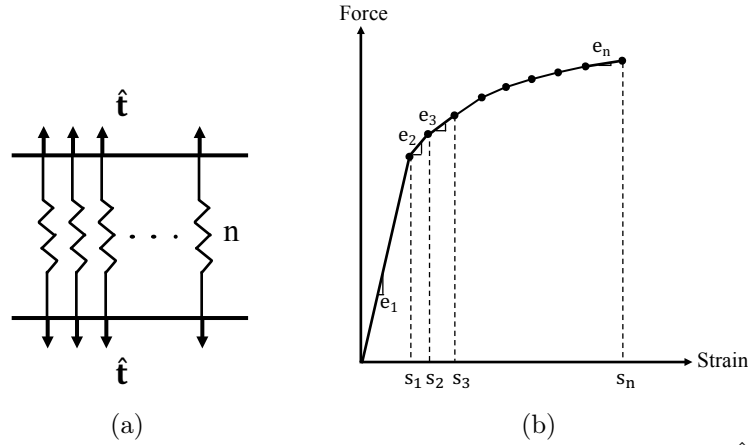


Figure 5.2: (a) Sub-layers of a single peridynamic bond undergoing force  $\hat{\mathbf{t}}$ ; (b) Constructing  $n$  layers and recording their critical stretch values at which they yield.  $e_1$  is the Young's modulus ( $E$ ) of the material.

parallel (shown in Fig. 5.2(a)). They all undergo the same force density  $\mathbf{t}$  but each one yields at a different critical stretch value which can be calculated from the actual stress-strain (or force-displacement) curve of the material obtained from the uniaxial tensile experiment. The total force density in a bond is assumed to be

$$\hat{\mathbf{t}} = \sum_{k=1}^n W_k \hat{\mathbf{t}}_k \quad (5.1)$$

where  $W_k$  is the weighting factor of the  $k^{th}$  sub-layer and is given by

$$W_k = \frac{e_k - e_{k+1}}{E}, \quad e_{k+1} = 0 \quad (5.2)$$

where  $e_k$  is the slope of the  $k^{th}$  sub-layer and  $E$  is the Young's modulus and it is easy to show that  $\sum_{k=1}^n W_k = 1$ .

As the deformation occurs, each bond stretches until it reaches its first sublayer's critical yield stretch ( $s_1$ ) beyond which it yields. As the stretching process evolves, more sub-layers tend to yield resulting the whole bond to become weaker. Each time a sub-layer with a weighting factor of  $W$  yields, it has the same effect as if the bond force is increased by its lost weighting factor.

In peridynamics, local damage of a particle, denoted as  $\Phi_i$  is defined as the ratio of the number of broken bonds to the total number of bonds attached to particle  $i$ . Once a bond breaks, it will be permanently removed from all force density calculations. Damage index is a scalar and ranges from 0 (unbroken bond) to 1 (completely broken bond) and can be computed by (Silling & Askari, 2005)

$$\phi(\mathbf{x}_i, t) = 1 - \frac{\int_{\mathcal{H}} \mu(\mathbf{x}_j - \mathbf{x}_i, t) dV_j}{\int_{\mathcal{H}} dV_j} \quad (5.3)$$

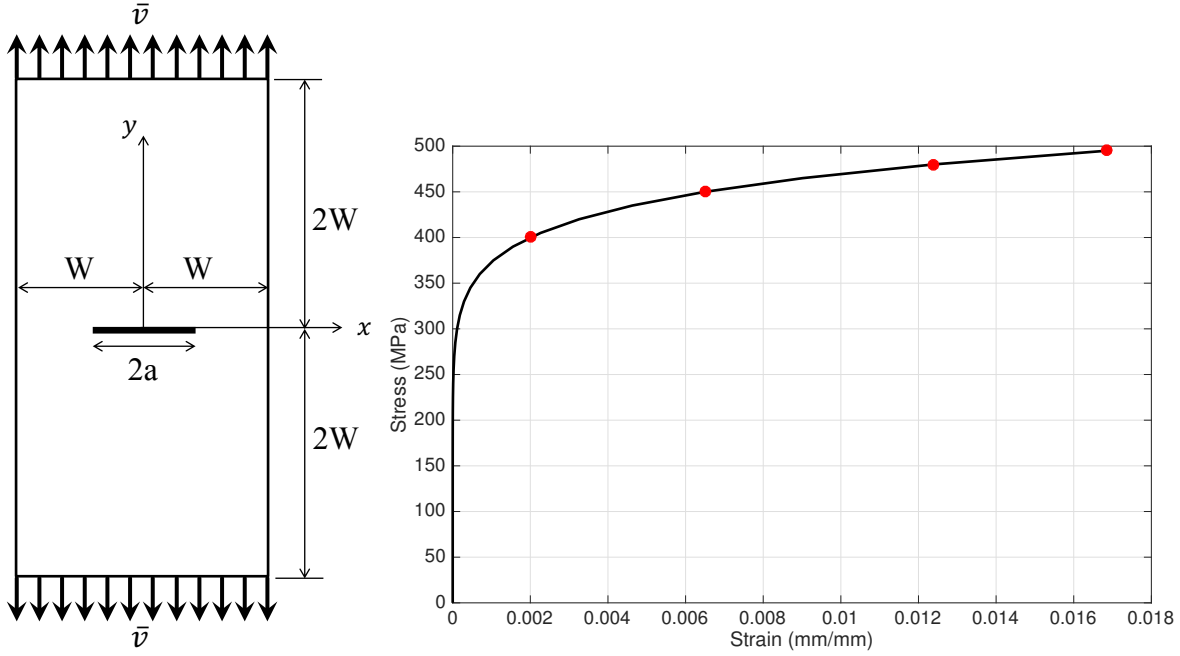


Figure 5.3: (a) A finite center-cracked plate with crack size of  $2a$ , and tensile displacement boundary condition made from ferrite steel; (b) Stress-strain curve for ferrite steel given in Eq. (5.7) and constructing sublayers with selected stress-strain pairs

where  $\mu$  is a scalar-valued function defined as

$$\mu(\mathbf{x}_j - \mathbf{x}_i, t) = \begin{cases} 0 & \text{if } s \geq s_{cr} \\ 1 & \text{if } s < s_{cr} \end{cases} \quad (5.4)$$

Similar to the damage index parameter, we can introduce a new parameter called *yield index* defined for a particle as the ratio of the evolved plastic deformation in the bonds attached to it divided by the total number of attached bonds. It can be represented as

$$\psi(\mathbf{x}_i, t) = 1 - \frac{\int_{\mathcal{H}} \lambda(\mathbf{x}_j - \mathbf{x}_i, t) dV_j}{\int_{\mathcal{H}} dV_j} \quad (5.5)$$

where  $\lambda$  is defined as

$$\lambda(\mathbf{x}_j - \mathbf{x}_i, t) = \begin{cases} 0 & \text{if } s > s_{yi} \\ 1 & \text{if } s \leq s_{y1} \end{cases} \quad \text{where } i = 1, 2, \dots, n \quad (5.6)$$

Yield index also ranges between 0 (unyielded bond) and 1 (yielded bond). Clearly, bond breaks once its stretch reaches  $s_{cr}$ . At that point, local damage  $\phi_i = 0$  and  $\psi = 1$ .

Table 5.1: Material properties of ferritic steel in center-cracked plate used by (Gullerud *et al.*, 2000)

| Material      | Elastic modulus (GPa) | Poisson's ratio | Density (g/cm <sup>3</sup> ) | $\sigma_0$ (MPa) | $\epsilon_0$ | $n$ |
|---------------|-----------------------|-----------------|------------------------------|------------------|--------------|-----|
| Ferrite steel | 200                   | 0.3             | 7800                         | 400              | 0.002        | 0.1 |

Table 5.2: Weighting factors and critical bond stretch values for yield prediction in compact tension specimen

|                        |                   |                   |                   |                   |
|------------------------|-------------------|-------------------|-------------------|-------------------|
| weighting factor       | $W_1 = 0.9444$    | $W_2 = 0.0301$    | $W_3 = 0.0087$    | $W_4 = 0.0168$    |
| critical yield stretch | $s_{y1} = 0.0020$ | $s_{y2} = 0.0065$ | $s_{y3} = 0.0124$ | $s_{y3} = 0.0168$ |

## 5.2 Benchmark problems

Analysis of conventional fracture mechanics problems is of substantial importance in examining the resistance to fracture in engineering materials. In this context, we present plastic deformation prediction using sublayer method for two benchmark problems in fracture mechanics and compare our results with those published in literature.

### 5.2.1 Center cracked plate in tension

The first benchmark problem models ductile crack growth in a center-cracked finite plate under a tensile displacement boundary condition. An experimental work done by (Gullerud *et al.*, 2000) on such a specimen is chosen here for validation. They have conducted their experiments on medium strength ferrite steel for which the properties are given in Table 5.1.

The stress-strain data obtained by (Gullerud *et al.*, 2000) was fitted to the following work-hardening equation

$$\frac{\epsilon}{\epsilon_0} = \left(\frac{\sigma}{\sigma_0}\right)^n \quad (5.7)$$

where  $\sigma_0$  and  $\epsilon_0$  are the reference (yield) stress and strain, and  $n$  is the work hardening exponent (Table 5.1). Figure 5.1 shows the plot of Eq. (5.7) for this type of steel along with selected four segments of sublayers between the yield point and the ultimate tensile stress. Weighting factors of sublayers and their critical yield stretch values are also measured from the graph and are tabulated in Table 5.2.

The geometry of the specimen is illustrated in Fig. 7.3. The initial crack to width ratio used here is  $2a/2W = 0.6$  with  $W = 50$  mm. The crack is modeled by removing the bonds shared between particles that are across it as previously suggested by (Madenci & Oterkus, 2014). The plate is discretized into a  $200 \times 400$  grid with uniform grid spacing of  $\Delta x = 0.5$  mm. In order to reproduce the experimental condition, we have monitored the force as a function of displacement and imposed it on a *fictitious layer* of width of horizon ( $\delta$ ) along the boundaries. (Macek & Silling, 2007) showed that adding such layer ensures that the assigned initial conditions will be properly distributed over the real particles of the

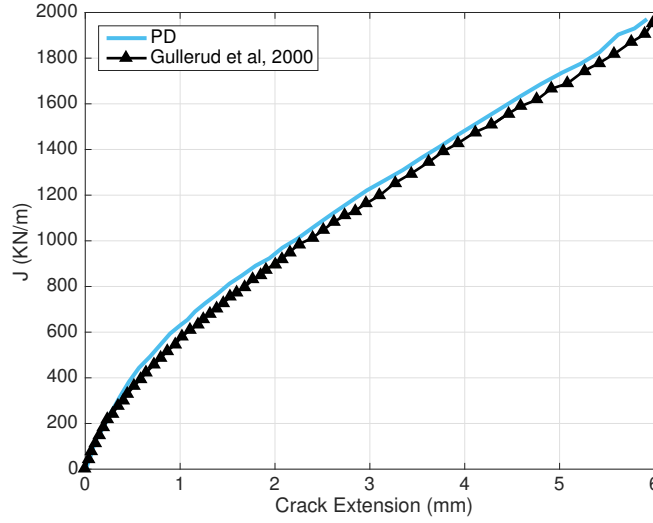


Figure 5.4: Comparison of PD J-integral vs. experimental results by (Gullerud *et al.*, 2000)

material. Hence, the total number of particles in this simulation is 81200 from which 1200 belong to the "fictitious" layer.

(Gullerud *et al.*, 2000) used computational cell methodology to model crack growth in different types of steel and evaluated their R-curves. Computational cell method which is originally presented by (Xia & Shih, 1995a; Xia & Shih, 1995b) proposes an engineering approach to evaluate R-curve by defining computational cell elements over the region of expected crack growth through involving the micro-mechanisms that lead to crack propagation. Therefore, computational cell method does not provide any information regarding the regions of the material which are further from the crack tip, yet might be susceptible to fracture due to being weaker (ex.along the boundaries) or due to the presence of other types of defects. Figure (5.4) shows that a good agreement is achieved between the computed values of PD J-integral using 2.29 combined with sublayer method and the J-integral values measured by (Gullerud *et al.*, 2000) using computational cells.

## Unloading

In order to assure that the presented model is capable of predicting unloading, a single random PD bond is chosen and its stretch and bond force are tracked. Due to the very non-linear behavior of PD force, Fig. 5.5 looks very noisy, yet it shows the correct trend for loading and unloading as the deformation continues.

Initiation and evolution of plastic deformation in the center-cracked specimen is visualized here through the contours of displacement, velocity, and yield index. Figures (5.6) and (5.7) show the vertical displacement and velocity contour, respectively as crack opens more and more due to the applied displacement.

Yield index per particles is illustrated in Fig. (5.8). As expected, crack growth initiates



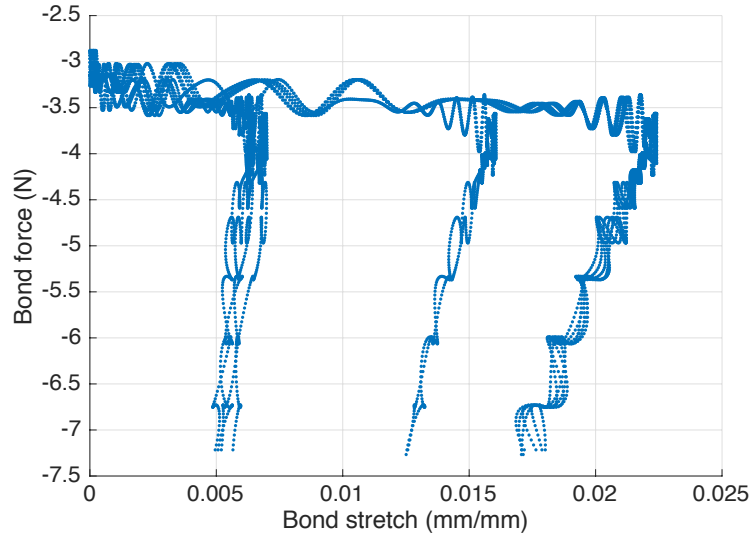


Figure 5.5: Bond force vs. stretch for a single bond undergoing loading and unloading

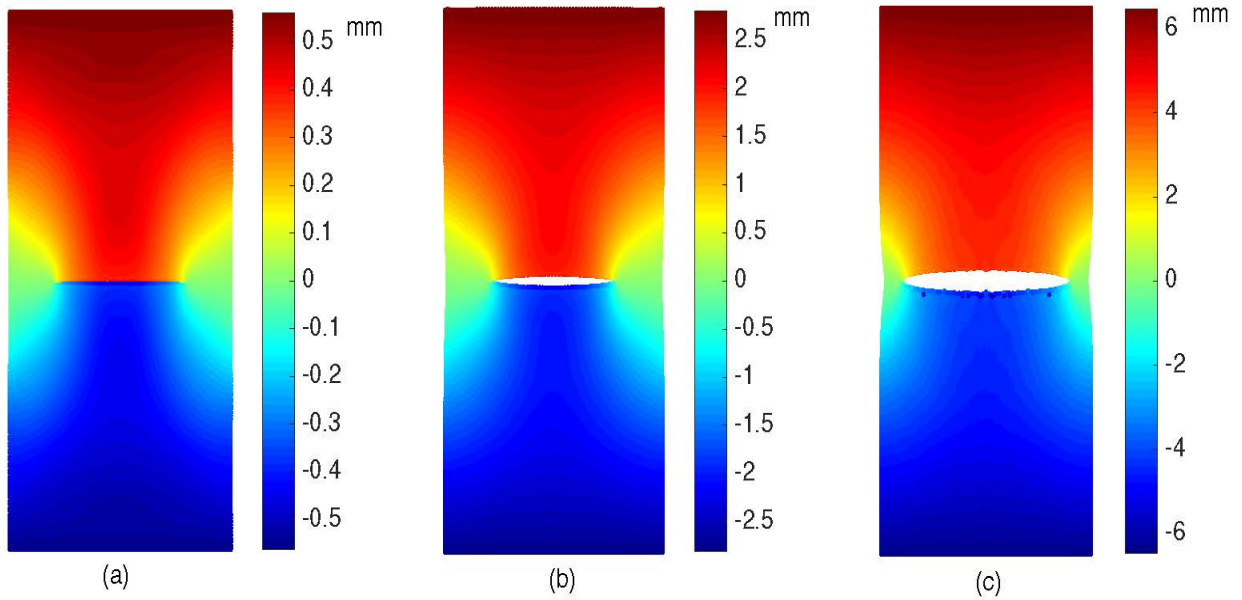


Figure 5.6: Subsequent displacement contours in the  $y$ -direction for (a)  $\bar{v} = 0.704$  mm, (b)  $\bar{v} = 0.985$  mm, (c)  $\bar{v} = 6.471$  mm

from regions with high stress concentration at crack tips. Fig. (5.8b,c) shows the dissipation of yield through the material under larger displacements.

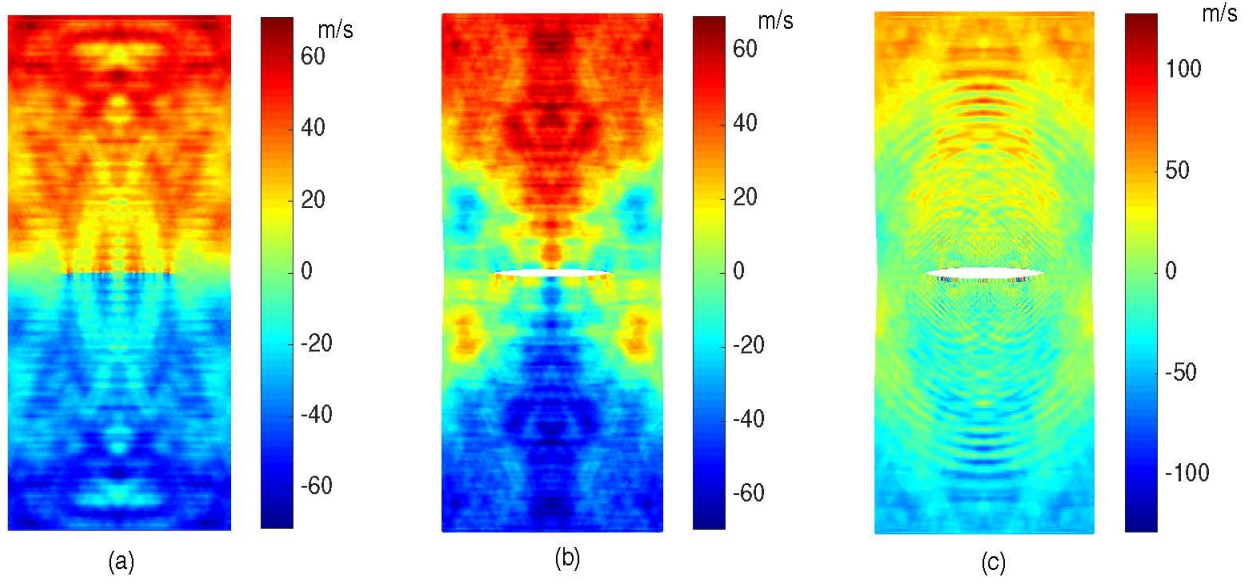


Figure 5.7: Subsequent velocity contours in the  $y$ -direction for (a)  $\bar{v} = 1.270$  mm, (b)  $\bar{v} = 2.813$  mm, (c)  $\bar{v} = 3.871$  mm

Table 5.3: Material properties of compact tension specimen reported in (Areias *et al.*, 2014)

| Elastic modulus (GPa) | Poisson's ratio | Density (g/cm <sup>3</sup> ) | $\sigma_0$ (MPa) |
|-----------------------|-----------------|------------------------------|------------------|
| 210                   | 0.3             | 8050                         | 445              |

### 5.2.2 Compact tension

The second benchmark problem is modeling the compact tension (CT) problem which is depicted in Fig. (5.9a) in two-dimensions. The domain is discretized into a  $200 \times 192$  grid with  $\Delta x = 0.3125$  mm. The relevant data including the geometry, stress-strain curve and material properties are adapted from a computational and experimental work performed by (Areias *et al.*, 2014) and (Samal *et al.*, 2009). It undergoes a tensile loading which is applied as a function of displacement in our simulation to the "fictitious" material points located inside the holes but are obviously eliminated from PD force calculation process. Therefore, total number of particles is 37724 from which 2316 are fictitious. Figure 5.9 shows the stress-strain curve of the CT specimen annotated with the 6 sublayers. The weighting factors and critical yield stretch values are computed in Table 5.4.

R-curve of the material is the parameter that is used to validate our model. Figure 5.10 shows the comparison between J-integral obtained from experimental and a mesh-independent finite element approach by (Samal *et al.*, 2009). Peridynamics seem to have a better agreement with the FEM results. However, it provides the exact same trend as was reported in the experiment by (Samal *et al.*, 2009).

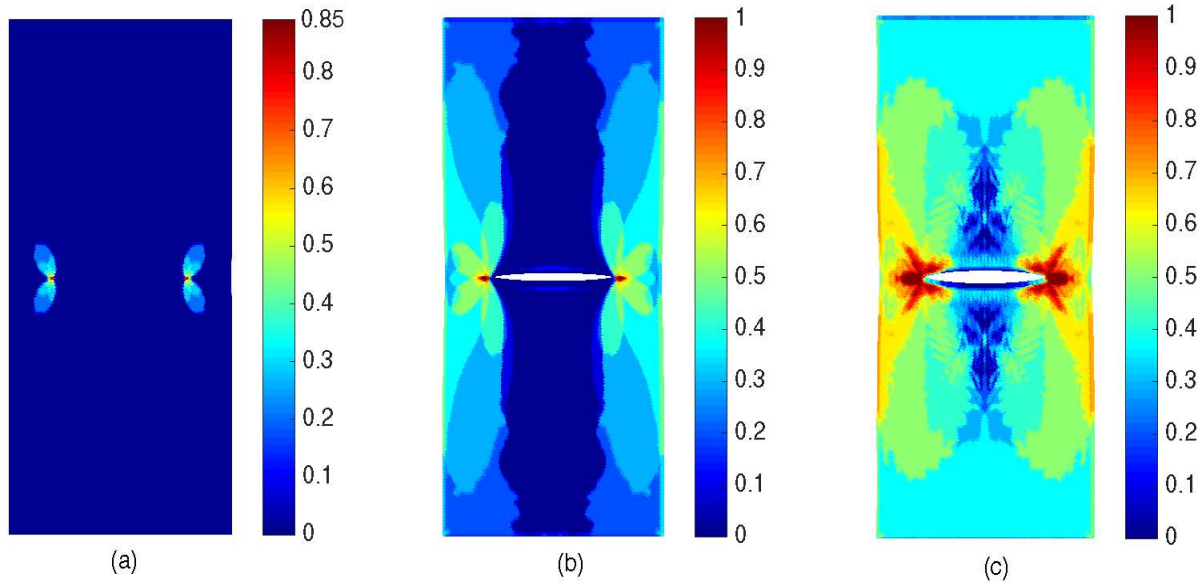


Figure 5.8: Subsequent yield index contours for (a)  $\bar{v} = 1.270$  mm, (b)  $\bar{v} = 2.813$  mm, (c)  $\bar{v} = 4.572$  mm

Table 5.4: Weighting factors and critical bond stretch values for yield prediction in center-cracked plate

|                          |                                  |                                  |                                  |
|--------------------------|----------------------------------|----------------------------------|----------------------------------|
| Weighting factors        | $W_1 = 8.9846 \times 10^{-1}$    | $W_2 = 4.9130 \times 10^{-2}$    | $W_3 = 1.9144 \times 10^{-2}$    |
|                          | $W_4 = 6.6750 \times 10^{-3}$    | $W_5 = 7.1325 \times 10^{-3}$    | $W_6 = 1.9460 \times 10^{-2}$    |
| Critical yield stretches | $s_{y1} = 5.6800 \times 10^{-2}$ | $s_{y2} = 1.207 \times 10^{-1}$  | $s_{y3} = 2.8640 \times 10^{-1}$ |
|                          | $s_{y4} = 5.2945 \times 10^{-1}$ | $s_{y5} = 8.2600 \times 10^{-1}$ | $s_{y6} = 1.1030$                |

Figure 5.11 shows the contour plots for the yield index as the plastic deformation evolves through the material. It can be easily observed that PD does not only predict the crack growth path in front of it correctly but also it highlights all the regions with high stress intensity factors spontaneously. As opposed to the conventional continuum theory which has to be provided with a pre-defect, in peridynamics, damage and yield phenomena are part of the solutions.

We monitor the force  $F$  as a function of the imposed displacement  $v$  and compare with the experimental results reported in (Samal *et al.*, 2009). This comparison is presented in Fig. 18 where good agreement can be observed, despite the slightly higher values of reaction obtained here. Note that higher numerical values were also reported by (Samal *et al.*, 2009).

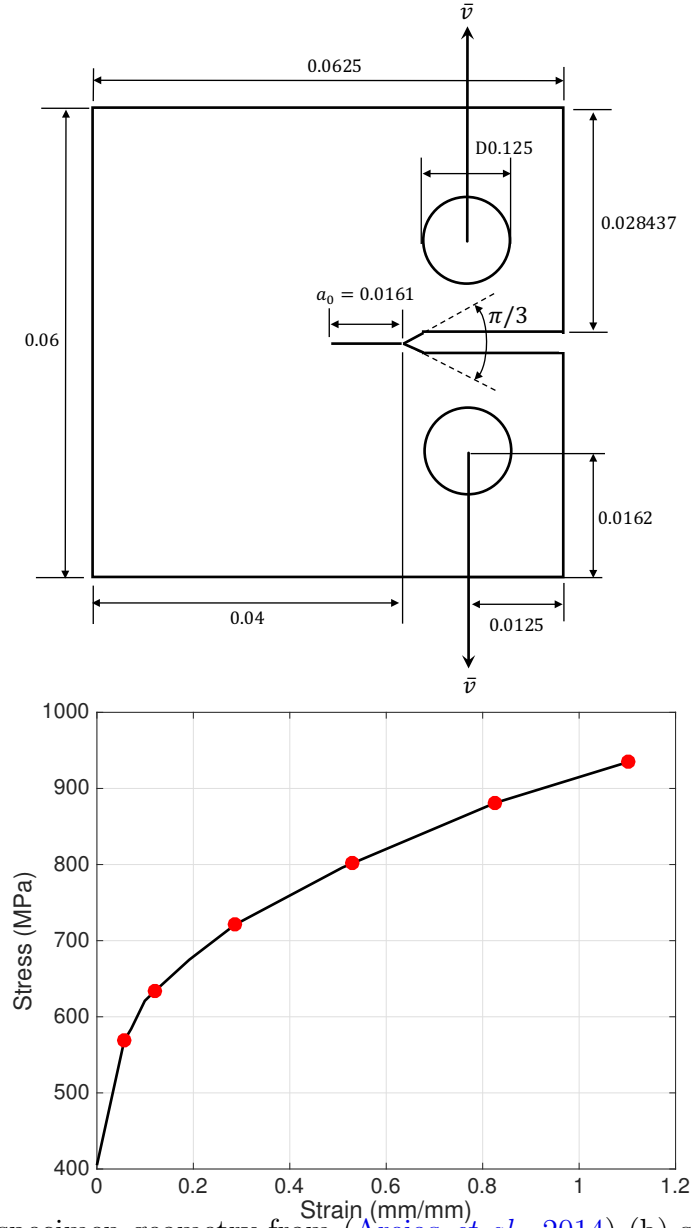


Figure 5.9: (a) CT specimen geometry from (Areias *et al.*, 2014) (b) stress-strain curve for the material used in the CT specimen from (Areias *et al.*, 2014)

### 5.2.3 Summary

In this chapter we introduced a new approach to model plastic deformation with PD theory inspired by the mechanical sublayer method (Besseling, 1953). We presented how yielding can be modeled per PD bonds and defined a new metric called yield index for measuring local irreversible damage caused by plastic deformation. We validated our PD approach

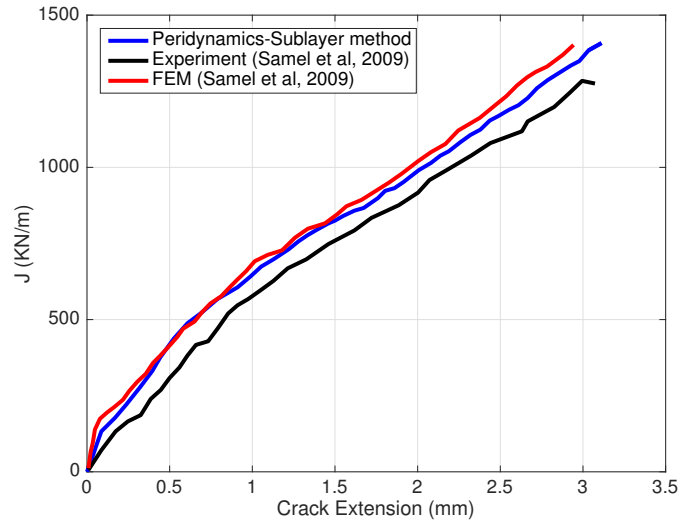
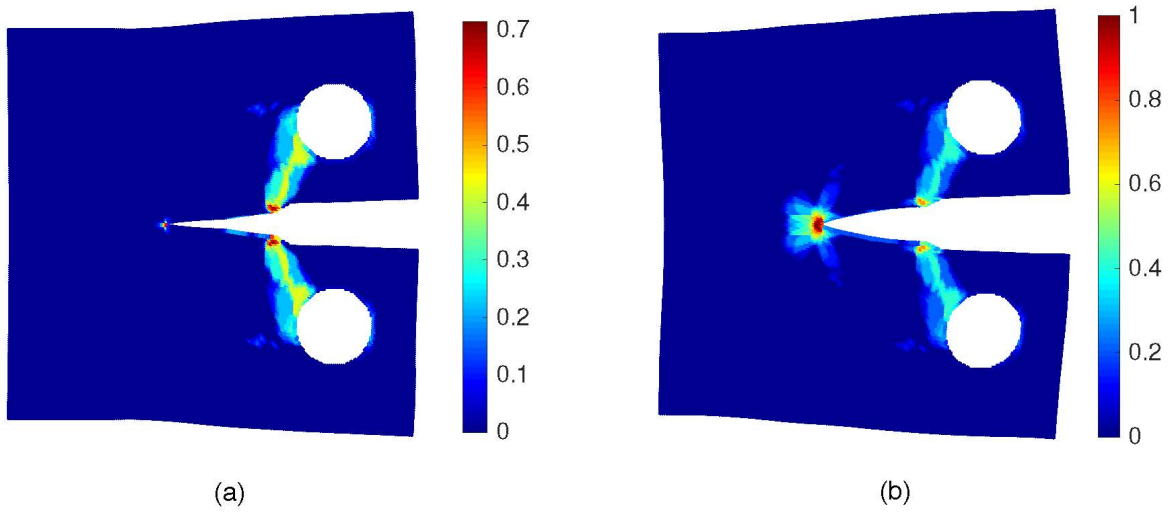


Figure 5.10: J-integral calculation comparison for the CT specimen

Figure 5.11: Yield index contour for the CT specimen for (a)  $\bar{v} = 1.99$  mm and (b)  $\bar{v} = 2.82$  mm

using experimental and analytical results for standard benchmarks including center-cracked plate and compact tension problem.

## Chapter 6

# Peridynamic for Ballistic Impact and Body Armor Design

### 6.1 Introduction

This chapter considers using peridynamic modeling to study contact between two bodies due to an impact event. Specifically, we focus on the ballistic impact that occurs in body armor systems. Researchers have explored modeling dynamic fracture and impact with PD theory but modeling this specific application has not been investigated before. In this chapter, we aim to propose two body armor systems that minimize ballistic impact effects as well as being light-weight and comfortable to wear. We perform an impact analysis by computing the velocity profile of the projectile before being stopped by the armor system. Evaluation is based on how much the bullet penetrates into the proposed design structure and how fast it can be stopped. We study common bullets with available public information.

#### 6.1.1 Body armor

The primary objective of body armor research is to develop a low-cost, light-weight, secure garment system of high ballistic impact resistance. Ballistic protection is a strong function of the material impact resistance. The ductile armor component absorbs the impact energy of an impinging projectile in the form of plastic deformation, whereas the brittle armor component (outer layer) dissipates the kinetic energy of the projectile by micro-fracturing. Advanced ceramics used in various armor applications as outer layers have been extensively studied in the past few decades ([Wilkins \*et al.\*, \(1967\)](#); [Sternberg, 1989](#); [Sadanandan & Hetherington, \(1997\)](#)).

Until the beginning of the 20<sup>th</sup> century, body armors used to be designed as a monolithic plate of a single metal which was chosen based on its hardness. In the 1950s researchers proposed a better approach which was designing a dual-layer structure. The first layer facing the projectile was made of a very hard material and the second layer was very ductile to be able to absorb the kinetic energy of the projectile through plastic deformation. However, no other

criteria such as weight or mobility of the wearer was taken to account so they simply used two kinds of steel for two layers of their design. Despite the merits of this idea, this design did not become popular due to the heaviness of steel. In the late 1950s aluminum became widespread among ballistic protection manufacturers and the USA military was the first country in using them in armored vehicles. While the monolithic structure of aluminum was light, it was vulnerable to impacts caused by hard core (steel or tungsten) projectiles. In the early 1960s manufacturers switched back to the dual-layer philosophy and used high-hardness steel as the front layer backed by aluminum which resulted in a great weight reduction. In 1963, manufacturers came up with the revolutionary idea of using ceramics in the front layer backed by fiberglass reinforced composites which decreased weight further. Microcracking of the hard and brittle ceramic during impact leads to projectile blunting, reducing its penetration capability and, hence, the damage caused to the armor. Since that breakthrough, depending on the cost and technological advancement of manufacturing, different types of body armors were developed: for the hard front layer in ceramic armors boron carbide, silicon carbide, aluminum nitride or titanium diboride were used and for the backing panel several fiber-reinforced materials such as Aramid (Kevlar) and polyethylene were utilized. The low fracture toughness of the hard ceramic is compensated by these soft backing material of reinforced polymer. Another role of the backing material is to support post-impact fracturing of the ceramic plate and entrap the projectile. However, despite the high adsorption capacity of these high performance polymer fibers, approximately 20-50 layers of fabric are needed to meet safety regulations for typical projectiles ([Lee \*et al.\*, \(2003\)](#)), resulting in a bulky and stiff armor that is not comfortable to wear and limits the rapid response of the wearer.

Design efficiency and the total cost proportionally increase in body armors so the most effective armor is never economically reasonable to be manufactured. There is also a trade off between efficiency of the ballistic protector and weight. There are also safety requirements enforced by law which have to be met for different levels of protection. National Institute of Justice regulations of body armor standards require a maximum projectile penetration depth of less than 1.73 inches ([National standard-0101.04, \(2001\)](#)) to prevent blunt trauma of the wearer ([Bazhenov, \(1997\)](#)). The design of anti-ballistic structures is traditionally carried out empirically which relies on impact tests using the real projectile-target system in each test. This method, despite yielding definitive results, is prohibitively expensive and is not readily generalizable to all variations of different controlling factors of the problem such as impact velocity, projectile sizes and different material types, garment size and various material types. Therefore we believe the key ideas to design a body armor that has all the ideal properties are the following:

- Developing a parametric model that is fast and accurate to model variety of materials selections to chose the optimal one.
- Reinforcing the backing panel with directional fibers to increase the energy absorption during penetration of projectile.



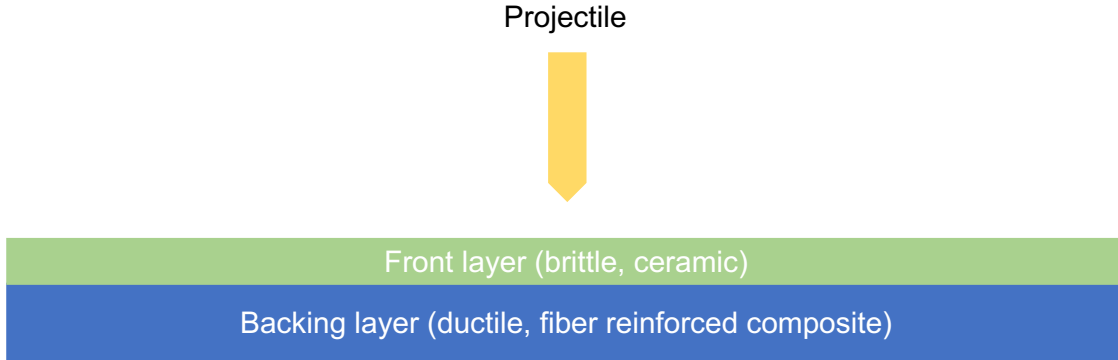


Figure 6.1: Schematic of the dual-material ballistic protection system

Designing and analyzing an optimal body armor requires a deep understanding of the damage in both layers as well as the advantages and disadvantages of the materials that could be used in both layers. Design choices must also be analyzed using a reliable framework that can model the system's behavior fast, accurate and easily. By far the best current framework to model dynamic fracture is Peridynamics. It was first introduced to model fracture and it has been proven to work as the best method for such purpose ([Madenci & Oterkus, 2014](#); [Silling, 2000](#)).

Figure 6.1 shows the schematic of a ballistic protection armor composed of the ceramic front layer backed by a ductile reinforced material.

## 6.2 Body armor design

In this section we propose to design body armor systems using PD modeling. We attack this problem in two attempts. We first assume that the bullet is rigid. This simplifies the computation because we do not need to model the impactor with its constructing particles. We use a bio-inspired structure by Abalone, a marine snail that protects itself with its convex shell with incredibly stiff layered structure. We analyze the performance of this structure and in our second attempt, we avoid this simplification and attack the problem in a more general way. We consider many possible material choices for the body armor as well as the bullet and try to find the combination that yields the best impact resistance.

### 6.2.1 Rigid impactor

Our first design choice is proposed to alter the classic design and instead use a bio-inspired composite structure for the baking polymer comprising an epoxy layered structure with strong interfaces between the layers. This microstructure mimics the abalone shell structure for which several studies of the unique mechanical properties (particularly, fracture toughness) have been reported in conjunction with manufacturing processes of producing



Table 6.1: Mechanical properties of materials used in the simulation

| Property                                           | SiC  | UHMWPE | WC-Co (projectile) |
|----------------------------------------------------|------|--------|--------------------|
| Elastic modulus (GPa)                              | 410  | 120    | 600                |
| Poisson's ratio                                    | 0.14 | 0.3    | 0.23               |
| Fracture toughness ( $\text{MPa}\sqrt{\text{m}}$ ) | 3.3  | 17.8   | 5.2                |
| Density ( $\text{kg/m}^3$ )                        | 3210 | 970    | 1500               |

such layered structures (Sarıkaya *et al.*, (1989); Lin & Meyers, (2005)). The intrinsic high strength and fracture toughness that of the abalone structure are attributed to the multiple interfaces of the laminated structure that play the role of energy absorption sinks. Layer interfaces can reflect back the stress waves generated by the penetrating projectile to decrease the driving force of crack propagation and increase the rate of energy absorption. The ballistic impact performance of the proposed model will be compared to that of a typical current armor system of dimensions  $20 \times 20 \times 7$  mm for a projectile consisting of WC-Co having a diameter of 9 mm, effective length of 2 mm, and blunt conical head with 5 mm apex diameter. In preliminary simulations, the impact velocity will be set equal to 100 m/s and the thickness of the SiC and polyethylene layers equal to 3.5 mm. The material properties of the armor components used in the present model are given in Table 6.1.

The ballistic performance depends on the kinetic energy loss of the projectile (Medvedovski, 2010). A decrease in projectile mass and/or velocity is the main goal of a secure garment system. Projectile mass reduction mainly occurs in the ceramic plate as a result of intense shattering and powdering, causing projectile blunting. Projectile deceleration mainly occurs in the polymeric backing not only because of projectile blunting but also due to particle attachment to the projectile apex that increase the friction with the backing material.

Peridynamic simulations illustrating the ballistic performance of the proposed composite armor material are included here to demonstrate the capability of the proposed peridynamics approach and potential of new armor design.

Figure 6.2 shows the initial problem setup and the final deformed structures of the conventional and proposed armor structure after 70  $\mu\text{s}$  and 50  $\mu\text{s}$ , respectively. Even though both structures successfully prevented full penetration by the projectile, the deformation of the monolithic polymer backing is significantly more pronounced than that of the layered polymer backing. This is attributed to the absorption of impact energy by the interfaces of the layered backing. Figure 6.3 shows the reduction of bullet's velocity over time as it penetrated through the proposed structure.

In addition to a qualitative comparison of the damaged patterns of the two models (Fig.

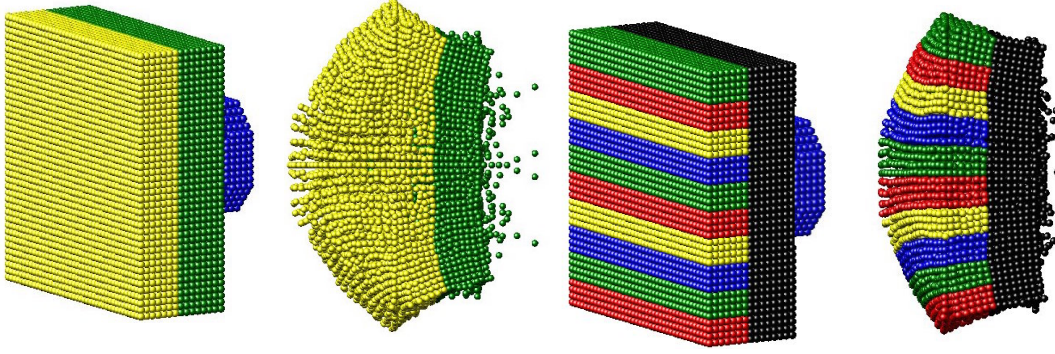


Figure 6.2: From left to right: 1) Initial configuration of WC-Co projectile (blue) and conventional body armor consisting of a SiC outer layer (green) and a homogeneous polyethylene backing (yellow), 2) deformed conventional body armor with buried projectile, 3) initial configuration of WC-Co projectile (blue) and new body armor design with a SiC outer layer (black) and epoxy-layered backing with strong interfaces (multi-colored), and 4) deformed new body armor with buried projectile.

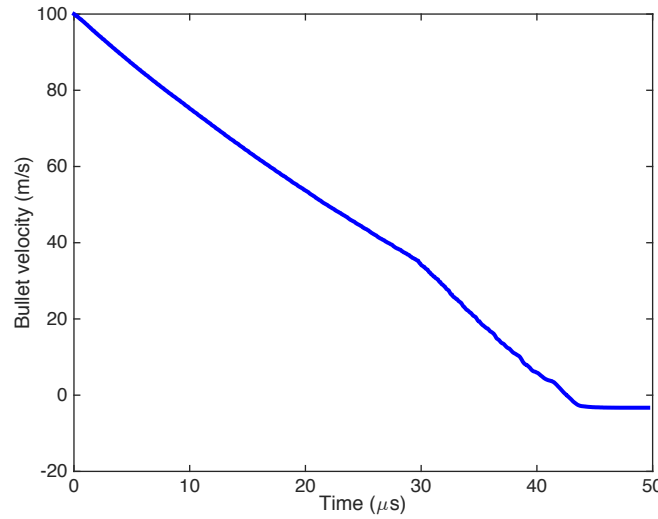


Figure 6.3: Bullet's velocity profile during the penetration

6.2), the projectile mass loss and velocity decrease for a fixed time after initial impact (i.e.,  $25 \mu\text{s}$ ) can be used to assess the penetration resistance of each design. In peridynamics, failure is predicted by tracking the position and velocity of all particles at each time step. Thus, the average velocity of the particles comprising the projectile represents the overall projectile velocity. Projectile mass loss is obtained by subtracting the mass of projectile particles with broken bonds from the total projectile mass. For a fixed time from initial impact, projectile mass loss (mainly occurring in the ceramic plate) is slightly higher in the new armor design; however, projectile deceleration is much more pronounced in the new design (Table. 6.2).

Table 6.2: Comparison between the mass and velocity reduction of the projectile in two models

| Backing plate        | Mass reduction | Velocity reduction |
|----------------------|----------------|--------------------|
| Monolithic UHMWPE    | 8.8 %          | 39.3 %             |
| Multi-layered UHMWPE | 10.4 %         | 56.8 %             |

### 6.2.2 Non-rigid impactor

We exploit our recently developed new PD framework for analyzing plastic and viscoplastic behavior of materials described in chapter 5 to model the backing panel behavior in maximizing the energy absorption while being cost and weight efficient. Using this plastic PD model, we also intend to analyze the composite structure of the backing panel by examining different unidirectional and multidirectional fiber reinforced composites. Our analysis will be specific to the type of projectile for which we are designing armor, such as handgun bullets, rifle bullets and heavy bullets.

Our goal is to develop a parametric PD model as the solution to choose the best effective ballistic protection body armor, following the program:

1. Exploiting our new viscoplastic PD model to study the behavior of the backing panel in the garment, summarized as follows:

We have adopted the idea of the sub-layer method (Besseling, 1953; Pian, 1987; Zienkiewicz *et al.*, 1973) to develop a new peridynamic framework to predict the behavior of a ductile material once it experiences loadings above its elastic limit. In this method, the key idea is that a peridynamic bond is assumed to be composed of  $n$  sub-layers which are in parallel. They all undergo the same force density  $\mathbf{t}$  but each one yields at a different critical stretch value which can be calculated from the actual stress-strain (or force-displacement) curve of the material obtained from the uniaxial tensile experiment.

The total force density in a bond is assumed to be a weighted sum of the force in all of the sub-elements. As the deformation occurs, each bond stretches until it reaches its first sublayer's critical yield stretch ( $s_1$ ) beyond which it yields. As the stretching process evolves, more sub-layers tend to yield resulting the whole bond to become weaker. Each time a sub-layer with a weighting factor of associated with its sub-element yields, it has the same effect as if the bond force is increased by its lost weighting factor. In order to model the viscoplastic behavior rate effects can be added to this plasticity model with little difficulty by accommodating the growing of the yield

forces with increased stretch rate by using the Cowper-Symonds relation :

$$\hat{\mathbf{t}}_k^{\hat{d}y} = \hat{\mathbf{t}}_k^y \left( 1 + \left( \frac{\dot{s}}{d} \right)^{1/p} \right)$$

where  $d$  and  $p$  are positive constants and  $\dot{s}$  is time derivative of the stretch in bond.

2. Using the PD formulation for non-isotropic materials (Bobaru & Silling, 2004; Askari *et al.*, 2006) to be able to optimize the design of the backing panel of composite with a variety of fiber orientations and propoerties.
3. We have implemented our PD model with a custom-made FORTRAN code which is highly optimized. Peridynamics, similar to any other particle method from the computational point of view, is highly parallelizable which provides us a good scalability of the solution. Our ultimate goal in this project is to parallelize our parametric model to be able to try different designing patterns quickly and accurately.

### 6.3 Performance analysis

Feasibility of the proposed research is initially investigated through analyzing several dual-material ballistic protection structures using different materials listed in table 6.3 using our new viscoplastic PD model which predicts the damage patterns due to the impact of bullets of different kinds.

We analyzed some possible combinations of the most widely used materials for projectiles, and for front and backing layers listed in Table 6.3 with their given material properties listed in the Table 6.4.

We first considered a fixed armor system of dimensions  $15 \text{ cm} \times 10 \text{ cm} \times 1 \text{ cm}$  with a cylindrical projectile with 5 cm diameter, and 2 cm effective length, and a blunt conical head with 2 mm appex diameter. In preliminary simulations, the impact velocity was set equal to 100 m/s and the thickness of the ceramic-based front layer is 2 mm and polymeric backing layer is equal to 8 mm. We ran the simulation for design patterns #4, #6, #11 given in Table 6.3. Figure 6.4 shows how projectile's velocity reduces from 100 m/s to 0 inside the body armor over the entire penetration process. Nearly all the plots show a plateau after approximately 70 100  $\mu\text{s}$  where the bullet enters the second layer of the armor in which 60% of its velocity will be decreased. By plotting bullet velocity versus time for all the suggested patterns one can find the most effective design for halting the projectile, taking account various other relevant factors. For example design #6 among the three chosen patterns was found to be the best performing but it also corresponds to the design that has the greatest weight.

Figure 6.5 shows the displacement contour along the projectile's direction through the structure for design pattern #11 after 50  $\mu\text{s}$ , 75  $\mu\text{s}$  and 375  $\mu\text{s}$ . This figure not only shows how the armor structure was able to stop the bullet but also it is able to demonstrate how the

Table 6.3: Possible combinations of three different projectiles with two different ceramic-based front layers and polymeric backing layers.

| Design # | Projectile material             | Front layer material | Backing layer material |
|----------|---------------------------------|----------------------|------------------------|
| #1       | Copper                          | $Al_2O_3$            | UHMWPE (Spectra)       |
| #2       | Copper                          | $Al_2O_3$            | Aramid (Kevlar)        |
| #3       | Copper                          | SiC                  | UHMWPE (Spectra)       |
| #4       | Copper                          | SiC                  | Aramid (Kevlar)        |
| #5       | Steel                           | $Al_2O_3$            | UHMWPE (Spectra)       |
| #6       | Steel                           | $Al_2O_3$            | Aramid (Kevlar)        |
| #7       | Steel                           | SiC                  | UHMWPE (Spectra)       |
| #8       | Steel                           | SiC                  | Aramid (Kevlar)        |
| #9       | Tungsten carbide-cobalt (WC-Co) | $Al_2O_3$            | UHMWPE (Spectra)       |
| #10      | Tungsten carbide-cobalt (WC-Co) | $Al_2O_3$            | Aramid (Kevlar)        |
| #11      | Tungsten carbide-cobalt (WC-Co) | SiC                  | UHMWPE (Spectra)       |
| #12      | Tungsten carbide-cobalt (WC-Co) | SiC                  | Aramid (Kevlar)        |

Table 6.4: Material properties used in simulating front layer, backing layer and the projectile

| Material / Properties | Elastic modulus (GPa) | Poisson's ratio | Density (kg/m <sup>3</sup> ) | Fracture toughness (MPa $\sqrt{m}$ ) |
|-----------------------|-----------------------|-----------------|------------------------------|--------------------------------------|
| SiC                   | 410                   | 0.14            | 3210                         | 3.3                                  |
| $Al_2O_3$             | 300                   | 0.21            | 3900                         | 3.5                                  |
| UHMWPE (Spectra)      | 120                   | 0.30            | 970                          | 18                                   |
| Aramid (kevlar 29)    | 70                    | 0.36            | 1450                         | 14                                   |
| Steel                 | 210                   | 0.30            | 7800                         | 80                                   |
| Copper                | 117                   | 0.35            | 8900                         | 20                                   |
| WC-Co                 | 600                   | 0.23            | 1500                         | 5.2                                  |

structure absorbs the kinetic energy of the bullet in part (b) and deforms back as unloading occurs when the bullet stops in part (c).

We will define the performance based upon key parameters in our study such as number and thickness of layers in the backing layer, armor weight and cost, energy absorption of the second layer, shape of the fibers used to reinforce the backing polymer, etc.

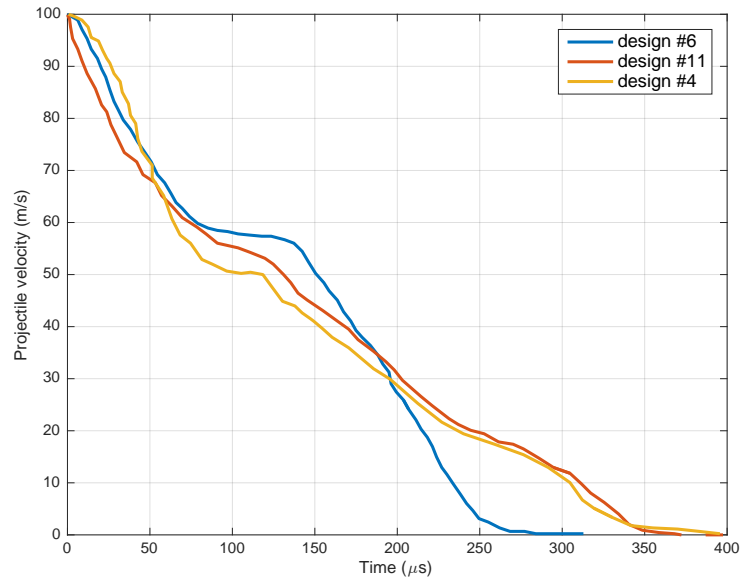


Figure 6.4: Velocity profile of different projectiles from Table 6.3

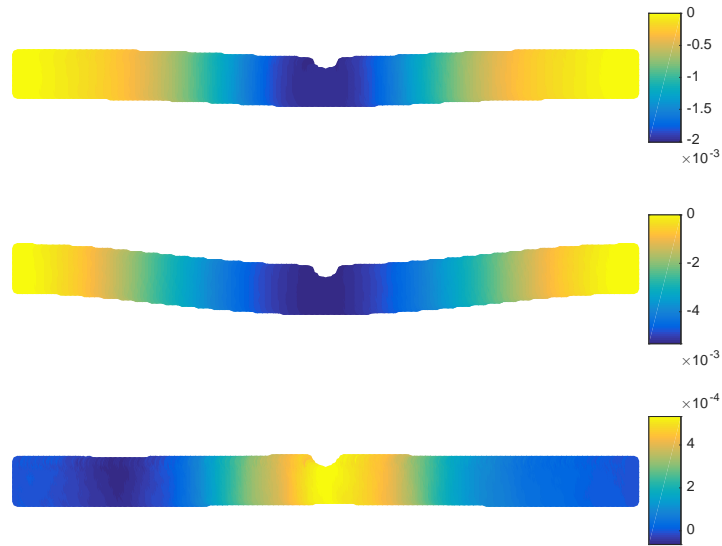


Figure 6.5: Displacement contour in vertical direction for design pattern #11. (a) bullet exits the front layer and penetrates thoroughly into it after  $50 \mu s$  (b) The second layer successfully envelops the bullet by absorbing its kinetic energy after  $75 \mu s$  (c) Unloading occurs and the bullet is pushed backward by small amount after  $375 \mu s$

## Chapter 7

# Parallel Algorithms for Peridynamic

### 7.1 Peridynamic is massively parallelizable

Peridynamics, similar to any other particle-based methods, requires computing interactions of many particles within the same *family*. This procedure can be computationally expensive if it is done using a sequential processor. However, PD is highly parallelizable due to the fact that interactions within each *family* are totally independent from another and therefore, parallel algorithms can be leveraged to reduce the number of computations drastically. Nevertheless, existing parallel codes for PD are either not available to the public domain (such as EMU) or limited to one version of the PD formulation such as Peridigm software (Parks *et al.*, 2012) which only covers the second formulation introduced in ???. In this thesis, we have used the first formulation for which there exists no parallel algorithm available and this is the first work on this matter. We use C++ for efficiency and speed purposes.

In this chapter we investigate the parallelization of particle simulation in peridynamics on distributed and shared memory CPU systems using MPI and OpenMP, respectively. We also implement the simulation on GPU using CUDA. We have built our parallel algorithms based on our modified version of the sequential program for the same problem. Results for weak and strong scaling efficiency are compared against each other and scaling performances for each model are evaluated and discussed. We show that these techniques can effectively speed up the simulation. CUDA implementation is shown to be the most powerful tool for massively parallel peridynamic simulations.

In many disciplines such as physics, astronomy and mechanics, researchers often study *dynamical systems* of particles. As a common fact among these and PD theory, each particle in such a system has a state at each time step and the system contains a set of equations that compute the state of each particle for the next time step. A natural question is to ask what the particles' states are after  $t$  time steps. It is known that such question is computationally hard (Reif & Tate, 1993); this means that it is unlikely that there exists an efficient algorithm



<sup>1</sup> that solves the problem. Thus, the widely used solution is to simply simulate the particles in each time step. This brings us to the central goal of this work: to simulate dynamical systems of particles as efficiently as possible.

It should be noted that the algorithms designed in this work are not merely designed for PD framework, and can be also used in any particle based method in which particles interact with each other through physical laws. More precisely, in such dynamical system of interest, each particle's state consists of its current position and velocity, both in two dimensional space. For each time step, each particle's position is changed according to its velocity whereas the velocity is changed corresponding to forces exerted by other particles within a specified cutoff distance, *horizon*, from the particle.

The trivial algorithm, in which one computes interactions between every pair particle in each time step, takes  $O(n^2)$  time per time step where  $n$  is the number of particles. This, however, is not the best one can do with this particular system. More specifically, the nature of the PD system allows us to check for interactions only between each particle and its family members within the horizon distance to it, That is, if we partition space into squares (or bins) each with height equal to the horizon, then it is enough for us to compute forces between each particle and the particles in its bin and the surrounding bins. When the density of the particle is constant, the expected number of particles in each grid is also constant, meaning that this “binning” algorithm takes only  $O(n)$  time.

In addition to improvement in asymptotic running time from  $O(n^2)$  to  $O(n)$ , we can use parallelization to speed up the simulation even further. We explore several models of parallelization in this work, including shared memory model and distributed memory model using OpenMP and MPI, respectively. We also experiment with parallelization via a graphics processing unit (GPU) via NVIDIA CUDA.

We explain our algorithms for each model along with the optimization techniques we tried and how effective they are. Then, in Section 7.3, we discuss the performances of our codes and whether they match our theoretical expectations. Finally, in Section ??, we suggest some ideas that may further improve performances of our codes.

## 7.2 Algorithms and Implementations

In this section, we describe our algorithms for the different models of parallel programming and their implementation. This section is organized as follows. For each model, we have a subsection corresponding to more details of the algorithm as well as implementation details and optimization techniques we have tried and how well they worked.

The algorithms in this chapter mainly focus on the optimality of the PD as a mesh-free method, and not PD parameters in details. Therefore, the details of the PD force calculations and other technicalities such as surface correction factor, volume correction factor, bond

---

<sup>1</sup>More specifically, if  $PSPACE \neq P$ , then there is no algorithm with running time of a polynomial in  $\log t$ .



**Algorithm 2** NAIVE SERIAL ALGORITHM

---

```

1: Given:  $E, \rho, \delta, bc, \Delta x, \Delta t$ 
2: Initialize particles coordinates
3: Initialize displacements and velocities
4: for each time step  $t = 1$  to  $T$  do
5:   for each particle  $i = 1$  to  $n$  do
6:     for each particle  $j = 1$  to  $n$  do
7:        $\xi \leftarrow \mathbf{x}_j - \mathbf{x}_i$ 
8:        $\eta \leftarrow \mathbf{u}_j - \mathbf{u}_i$ 
9:       update  $scr, fac$ 
10:       $\mathbf{F}_i \leftarrow \mathbf{F}_i + bc \eta \frac{\eta - \xi}{\xi \Delta x^3 scr fac (\mathbf{x}_j + \mathbf{u}_j - \mathbf{x}_i - \mathbf{u}_i)}$ 
11:   for each particle  $i = 1$  to  $n$  do
12:      $\mathbf{u}_i, \mathbf{v}_i \leftarrow \text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$ 
1: procedure  $\text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$ 
2:    $\mathbf{a}_i \leftarrow \mathbf{F}_i / \rho$ 
3:    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{a}_i \Delta t$ 
4:    $\mathbf{u}_i \leftarrow \mathbf{u}_i + \mathbf{v}_i \Delta t$ 
5:   return  $\mathbf{u}_i, \mathbf{v}_i$ 

```

---

constant, critical stretch bond, etc., which were all discussed in ??, are not here. PD Force computation is only given in the algorithms at a high level.

### 7.2.1 Serial algorithm

First, let us discuss the naive serial algorithm. As shown in Algorithm 2, at each time step, the force interactions between each pair of particles are calculated and stored as PD force of each particle. Every particle is then *moved*, i.e., its velocity and displacement is subsequently updated accordingly to its acceleration.

By using the binning method (also known as spatial partitioning or discretization), we can split our domain into small bins (or partitions). Each bin is simply a square of size being the horizon limit. Each particle then merely interacts with its own surrounding bins. Therefore, to calculate the forces applied on each particle, we no longer need to iterate through all other particles and this reduces computational time drastically. In fact, if the density of the particles<sup>2</sup> is constant, then the number of particles in each bin is constant in expectation. Hence, the running time is linear in the number of particles. The pseudo-code for the algorithm is shown in Algorithm 3.

---

<sup>2</sup>The density of the particles in the number of particles per unit square of the plane.

**Algorithm 3**  $O(n)$  SERIAL ALGORITHM (SPATIAL PARTITIONING)

---

```

1: Given:  $E, \rho, \delta, bc, \Delta x, \Delta t$ 
2: Initialize particles coordinates
3: Initialize displacements and velocities
4: Initialize bins
5: for each particle  $i = 1$  to  $n$  do
6:   assign particle  $i$  to a bin according to its position
7: for each time step  $t = 1$  to  $T$  do
8:   for each particle  $i = 1$  to  $n$  do
9:     for each bin  $B$  nearby  $i$  do
10:      for each particle  $j$  in bin  $B$  do
11:         $\xi \leftarrow \mathbf{x}_j - \mathbf{x}_i$ 
12:         $\eta \leftarrow \mathbf{u}_j - \mathbf{u}_i$ 
13:        update  $scr, fac$ 
14:         $\mathbf{F}_i \leftarrow \mathbf{F}_i + bc \eta \frac{\eta - \xi}{\xi \Delta x^3 scr fac (\mathbf{x}_j + \mathbf{u}_j - \mathbf{x}_i - \mathbf{u}_i)}$ 
15:   for each particle  $i = 1$  to  $n$  do
16:      $\mathbf{u}_i, \mathbf{v}_i \leftarrow \text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$ 
17:     update particle  $i$ 's bin according to its new position

```

---

### 7.2.2 Serial implementation details

The data structures used for the algorithm are vectors in C++ Standard Library. More specifically, we declare one vector for each bin; the vector contains the indices of all the particles in the bin. When we update a particle's bin, we just remove its index from the previous bin's vector and push it into the current bin's vector. Below we list two optimization techniques that are common in particle-based methods and the reasons we did not keep them in the final implementation.

#### 1. Iteration through bins instead of particles

For the two loops that we iterate through particles (line 9 and 16 in the pseudo-code), we can instead iterate through bins first and, in that loop, iterate through particles in the bin. It may not be clear from the pseudo-code what the benefit of such alternate looping is since updating a particle's bin is not expanded out. In the actual implementation, we need to go through the whole vector corresponding to the old bin to find the particle and remove it from the vector. On the other hand, if we use the alternate looping, we can directly delete it from the vector. Theoretically speaking the latter should be better. However, based on our experiment, looping by particles outperforms looping by bins. As a result, we keep the former in our final code. A reasonable explanation is that the more layer of the loops may add too much of an overhead when we loop by bins, which prevents us from seeing any improvement.

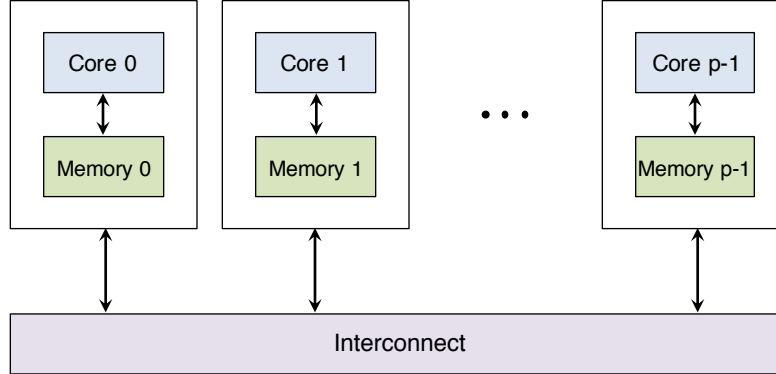


Figure 7.1: A distributed-memory system

## 2. Using smaller number of bins

While making each bin has side length the same as the horizon limit seems reasonable, the number of bins is  $\frac{\text{size}^2}{\text{horizon}^2}$ , which is  $3n$  in our case. While this is not too big, it seems that the memory usage overhead may worsen the performance of the code. We try changing decreasing the number of bins to smaller numbers. As we did so, we saw decline in performance so we kept the bin size to be the horizon distance.

### 7.2.3 Parallel algorithms

In the world of parallel multiple instruction, multiple data, or MIMD, computing systems can be categorized into two categories: distributed-memory and shared-memory systems. A distributed-memory system consists of a collection of core-memory pairs connected by a network, and the memory associated with a core is directly accessible only to that core as shown in Fig. 7.1. Whereas, a shared-memory system consists of a collection of cores connected to a globally accessible memory, in which each core can have access to any memory location, depicted in Fig. 7.2. We implement PD in both memory systems and discuss their benefits in the following sections.

Last PD parallel algorithm we propose in this work will be leveraging Graphical Processing Units (GPU). For that, we use CUDA as the programming interface used to write parallel code in heterogeneous architectures. We will discuss the main challenges that might occur when dealing with CUDA architecture as well as their benefits over the CPU-based parallelization techniques.

### 7.2.4 Shared-memory algorithm: OpenMP

OpenMP and Pthreads are two parallel programming models for shared memory parallelization schemes. In this work we have only implemented OpenMP. The pseudo-code 4 shows how the  $O(n)$  serial code is optimized with OpenMP.

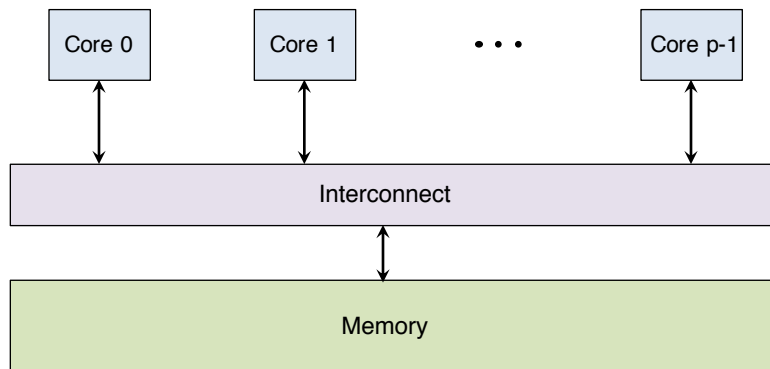


Figure 7.2: A shared-memory system

### 7.2.5 OpenMP implementation details

Our shared-memory code is almost exactly the same as that for serial except for the way *synchronization* is being handled. *Race condition* is a common issue in shared memory parallelization when multiple threads are updating the same shared variable simultaneously. In this problem, race condition may arise when two particles are removed from or added to the same bin when we are updating particles' bins. To prevent such race conditions, we have a lock (`omp_lock_t`) for each bin. Whenever the bin is modified, its corresponding lock is held by the thread that makes changes to the bin. While this may seem like excessive lockings on the surface, in practice, two threads rarely wants to modify the same bin due to the sparsity of the particles; in fact, as will be discussed below, this yields a better performance than other approaches that use less locks. Another trick which can be deployed is the *critical* directive for synchronization which simply causes each thread executes the *critical* region one at a time to check for minimum distance between each two particles. To reduce number of writes, saving the data is also assigned to the master thread. Here are two ways to reduce the number of required lockings:

1. We can split the large loops into two smaller loops: the first loop handles moving particles and removing it from the old bin, and the second loop adds the particles into the new bins. If the first loop is iterating over bins, then we can remove particles without using locks. Hence, we only need locks for adding particles and, intuitively, we should get some improvement. However, no improvements was observed and therefore, we concluded that the overhead of having two separate loops and having to iterate over bins instead of particles (see discussions in the serial section) weights in more than the improvement from less lockings.
2. Follow up from the previous item, if the large loop is divided into two smaller ones as stated above, then we can further reduce the number of locks required in adding particles to their new bins. Since most particles do not move too far from there previous bin, we can iterate through the bins, and, for each bin  $B$ , looks at the bins “close” to

---

**Algorithm 4** SHARED MEMORY (OPENMP) ALGORITHM

---

```

1: Given:  $E, \rho, \delta, bc, \Delta x, \Delta t$ 
2: Initialize particles coordinates
3: Initialize displacements and velocities
4: Initialize bins
5: Initialize locks ▷ We have one lock per bin
6: for each particle  $i = 1$  to  $n$  do
7:   assign particle  $i$  to a bin according to its position
8: for each time step  $t = 1$  to  $T$  do
9:   for each particle  $i = 1$  to  $n$  do ▷ This loop is executed in parallel
10:    for each bin  $B$  nearby  $i$  do
11:      for each particle  $j$  in bin  $B$  do
12:         $\xi \leftarrow \mathbf{x}_j - \mathbf{x}_i$ 
13:         $\eta \leftarrow \mathbf{u}_j - \mathbf{u}_i$ 
14:        update  $scr, fac$ 
15:         $\mathbf{F}_i \leftarrow \mathbf{F}_i + bc \eta \frac{\eta - \xi}{\xi \Delta x^3 scr fac (\mathbf{x}_j + \mathbf{u}_j - \mathbf{x}_i - \mathbf{u}_i)}$ 
16:   for each particle  $i = 1$  to  $n$  do ▷ This loop is executed in parallel
17:      $\mathbf{u}_i, \mathbf{v}_i \leftarrow \text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$ 
18:     if particle  $i$ 's move to a new bin then
19:       lock the old bin
20:       remove particle  $i$ 's from the old bin
21:       unlock the old bin
22:       lock the new bin
23:       add particle  $i$  to the new bin
24:       unlock the new bin

```

---

$B$  to see whether there are any particles that should be moved into  $B$ . This steps does not need any locks. Of course, there can be particles that move far away so we need to have another loop that takes care of that. We experiment with various distance that for “closeness” but never get any improvement over the original implementation.

### 7.2.6 Distributed memory algorithm: MPI

MPI is a parallel programming model for distributed memory parallelization. We also use binning for our MPI code. The central idea in our MPI implementation is that each processor is assigned with some bins that it is responsible for. At each time step, each processor computes the acceleration, velocity and displacement of the particles that belong to its bins and then sends the particles that leave its partitions to processors responsible for the destination partitions. Moreover, it sends to each processor the particles that are in bins adjacent to the bins responsible for this processor. These are needed to compute

**Algorithm 5** MPI ALGORITHM

---

```

1: Given:  $E, \rho, \delta, bc, \Delta x, \Delta t$ 
2: Initialize particles coordinates
3: Initialize displacements and velocities
4: Initialize bins
5: compute which processor is correspond to each bin
6: for all time steps  $t = 1$  to  $T$  do
7:   compute which processor(s) each particle this processor was responsible for in the
   previous step should be send to.
8:   send this processor's particles from previous step to other processors as necessary.
9:   receive particles from other processors.
10:  update this processor's bins.
11:  for each bin  $B$  that this processor is responsible do
12:    for each particle  $i = 1$  in  $B$  do
13:      for each bin  $B'$  nearby  $B$  do
14:        for each particle  $j$  in bin  $B'$  do
15:           $\xi \leftarrow \mathbf{x}_j - \mathbf{x}_i$ 
16:           $\eta \leftarrow \mathbf{u}_j - \mathbf{u}_i$ 
17:          update  $scr, fac$ 
18:           $\mathbf{F}_i \leftarrow \mathbf{F}_i + bc \eta \frac{\eta - \xi}{\xi \Delta x^3 scr fac (\mathbf{x}_j + \mathbf{u}_j - \mathbf{x}_i - \mathbf{u}_i)}$ 
19:  for each bin  $B$  that this processor is responsible do
20:    for each particle  $i = 1$  in  $B$  do
21:       $\mathbf{u}_i, \mathbf{v}_i \leftarrow \text{MOVE}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{F}_i)$ 

```

---

forces between particles correctly. Finally, it receives the particles that moved to its bins (or adjacent bins). The following pseudo-code shows the overall idea in the MPI algorithm.

### 7.2.7 MPI implementation details

We again have one vector for each bin. Each processor also records which bins it is corresponding to. As for communication, we use `Isend` when we send particles to other processors. For receiving, we use `Iprobe` to check whether there is any message ready to be received from any processor; if so, we receive the message using `Recv`. We use different tags for different time steps to ensure that we are not receiving particles for the future time steps. More discussions about our choice of communication protocols can be found below. Several optimization techniques we used are also listed below.

**Non-blocking vs blocking communication** We first used the (sometimes) blocking `Send`. This unfavorably increased the simulation time. It also makes the program to wait infinitely when the ordering of sends and receives are incorrect. We fix this by using `Isend`

instead, which complicates the implementation a bit since the buffer cannot be reused immediately. Nonetheless, it results in a huge improvement. Using `Iprobe` along with `Recv` is also beneficial and solves the problem of waiting for one specific processor to send messages while there are other messages from other processors for us to receive. On average we experienced near 5% improvement when `Iprobe` was used with `Recv`.

**Space partitioning.** We tried to assign partitions to processors in a way that decreases the communication complexity of our algorithm. In particular, we want to minimize the number of particles that need to be sent to multiple processors (because they are adjacent to partitions not belong to the processor it is assigned to). This is the same as trying to minimize perimeter of the partitioning of the space. We ended up using vertical stripes as our partition; each processor is responsible to a stripe. In our experiment, this works much better than a random partitioning. We also tried grid partition when the number of processors is four by dividing the plane into  $2 \times 2$  grid but we did not see any significant difference compared to the vertical stripes partitioning so we stick with the latter.

### 7.2.8 GPU: CUDA

Although increasing the CPU clock speed is certainly not the only method by which computing performance has been improved, it has always been a reliable source for improved performance. In recent years, however, manufacturers have been forced to look for alternatives to this traditional source of increased computational power. Because of various fundamental limitations in the fabrication of integrated circuits, it is no longer feasible to rely on upward-spiraling processor clock speeds as a means for extracting additional power from existing architectures. Because of power and heat restrictions as well as a rapidly approaching physical limit to transistor size, researchers and manufacturers have begun to look elsewhere.

In 2005, faced with an increasingly competitive marketplace and few alternatives, leading CPU manufacturers began offering processors with two computing cores instead of one. Over the following years, they followed this development with the release of three-, four-, six-, and eight-core central processor units. Sometimes referred to as the multicore revolution, this trend has marked a huge shift in the evolution of the consumer computing market.

In comparison to the central processor's traditional data processing pipeline, performing general-purpose computations on a GPU is a new concept. In fact, the GPU itself is relatively new compared to the computing field at large. However, the idea of computing on graphics processors dates back to late 80s.

Unlike previous generations that partitioned computing resources into vertex and pixel shaders, the CUDA Architecture included a unified shader pipeline, allowing each and every arithmetic logic unit (ALU) on the chip to be marshaled by a program intending to perform general-purpose computations. Because NVIDIA intended this new family of graphics processors to be used for general purpose computing, these ALUs were built to comply with IEEE requirements for single-precision floating-point arithmetic and were designed to use an

---

**Algorithm 6** CUDA ALGORITHM

---

```

1: initialize bins
2: copy the particles from CPU to the GPU
3: for each time step  $t = 1$  to  $T$  do
4:   assign particles to bins          ▷ Executed in GPU; inserting to a bin is atomically
   performed
5:   compute forces withing bins      ▷ This is executed in GPU
6:   moving particles                 ▷ This is executed in GPU
7: copy the particles from GPU back to the CPU

```

---

instruction set tailored for general computation rather than specifically for graphics. Furthermore, the execution units on the GPU were allowed arbitrary read and write access to memory as well as access to a software-managed cache known as shared memory. All of these features of the CUDA Architecture were added in order to create a GPU that would excel at computation in addition to performing well at traditional graphics tasks.

In this work, we use CUDA as another programming interface to implement PD in parallel. The idea is that the user is in control of the CPU (host) and the GPU (device). Most of the computational effort is done in the GPU since it is more amenable to parallelization, as long as not too much communication effort comes with it. CUDA is drastically powerful for massive parallelization speedups. However, it may not be suitable for medium scale parallelization because it needs memory management for devices. Since the different threads in the GPU share memory, our algorithm for this case is basically the same as in the shared-memory case up to implementation details of data structures for our data. The pseudo-code for our CUDA implementation is shown below.

### 7.2.9 CUDA implementation details

Since STL vector does not work on GPU, we simply allocate memory for all the bins before the simulation begins. We allocate a memory for six particles for each bin. We then simply use this array in place of the vectors. The other main difference between our CUDA code and others is that we always recompute the bins every time step whereas, in other models, we only move each particle to its new bin if its bin changes. This involves both removing and adding particles to bins, which requires locks. On the other hand, recomputing bins does not involve removing particles from bins, which helps us avoid locking the bins; here we can just use `atomicAdd` to calculate the end index in the bin that the particle is moved to.

We note that the reason for the number six used above is that, in the given code, the correctness is checked by checking whether the square of the minimum distance between two particles are at least 0.4 times the square of the cutoff distance. From this, we can prove that, if an implementation passes such test, then, at any time step, each bin contains at most six particles as follows. We first divide each bin into six rectangles with height cutoff/2 and



Table 7.1: Material properties and PD parameters of the center-cracked plate shown in Fig. 7.3

| Elastic modulus (GPa) | Poisson's ratio | Density (g/cm <sup>3</sup> ) | $\dot{u}_y(x, \pm L/2, t)$ | $s_{cr}$       | $\Delta t$              |
|-----------------------|-----------------|------------------------------|----------------------------|----------------|-------------------------|
| 192                   | 0.33            | 8000                         | $\pm 20$ m/s               | 1 (no failure) | $1.3367 \times 10^{-8}$ |

width cutoff/3. Each rectangle cannot contain more than two particles because otherwise the square of the distance between the two particles is at most  $(1/2)^2 + (1/3)^2 < 0.4$  times the square of the cutoff. As a result, there can be at most six particles in each bin, unless the checker given is incorrect.

**Iterating by bin instead of particles.** Similar to other implementations, we have a choice of looping over particles or looping over bins when we compute forces and move particles. We tried iterating over bins but its performance is slightly worse than iterating over particles so we use the latter in our final implementation.

## 7.3 Results and discussion

We now present the parallelizations results using the proposed algorithms. We keep the problem setup consistent across all the algorithms. It should be noted that we do not allow bond breaking to occur in this chapter so that the number of particles will remain the same at all time. However, we do account for particles moving around different horizons and changing their family members. This means that searching for family members is required at all time steps and this is the most rigorous way of solving a PD problem. Most work in the literature tend to assume particles do not change their family because it drastically reduces the computations. However, here we want to focus on finding the best parallel algorithm that works for PD even in the hardest possible case.

We have chosen the classic center-cracked plate problem under velocity loading boundary condition with specific material properties given in (Madenci & Oterkus, 2014). Figure 7.3 shows our setup where the square plate is of length  $L$  with center crack of length  $2s = 0.01m$ . The domain is discretized into  $500 \times 500$  unless otherwise is stated or a test is repeated for different number of particles.

The experiment run on TACC Stampede supercomputer with 6400 nodes configured with two Xeon E5-2680 processors and one Intel Xeon Phi SE10P Co-processor (TACC, 2015).

### 7.3.1 Speed up

We first show the results for the naive and optimized serial algorithms. A common way to show the Figure 7.4 shows a comparison between running time of the above algorithms for different number of particles (500-8000). For the naive code, slope of the data is 1.992 whereas for the optimized code it is 1.12 which matches our expectations.

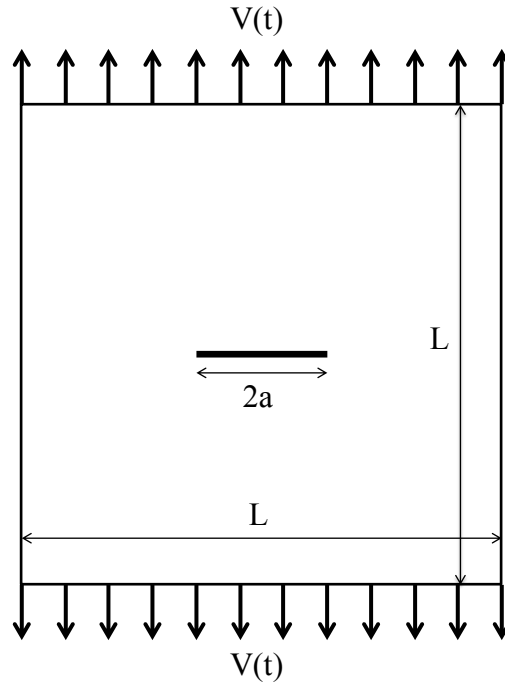


Figure 7.3: Center-cracked plate under velocity boundary condition for testing the parallel algorithms

Figure 7.5 shows the simulation time for different number of particles for fixed number of threads of 4 and 24, respectively. MPI jobs were performed on 3 machines and it is clear that an OpenMP implementation with 4 number of threads performs faster because it uses shared memory compared to MPI with overheads due to message-passing between 4 processors. However, MPI relatively scales better with 24 threads. The performance of the GPU code is also shown in this figure for the sake of comparison despite the fact that they were executed on different machines. Clearly CUDA speeds up the performance by a large scale using many thousands of threads.

Figure 7.6 gives a better insight into the performance of MPI and OpenMP. For a fixed number of particles (50000), it shows the simulation time in seconds. OpenMP does not scale beyond 8 threads while MPI continuously scale with increase in thread numbers. However, OpenMP outperforms MPI in terms of simulation time for the same number of threads.

Fig. 7.7 is another comparison between MPI and OpenMP in terms of the number of flops. While number of flops in MPI linearly increases with number of threads, OpenMP consistently remains as low as less than 250 Mflops/sec across all variations of number of threads.

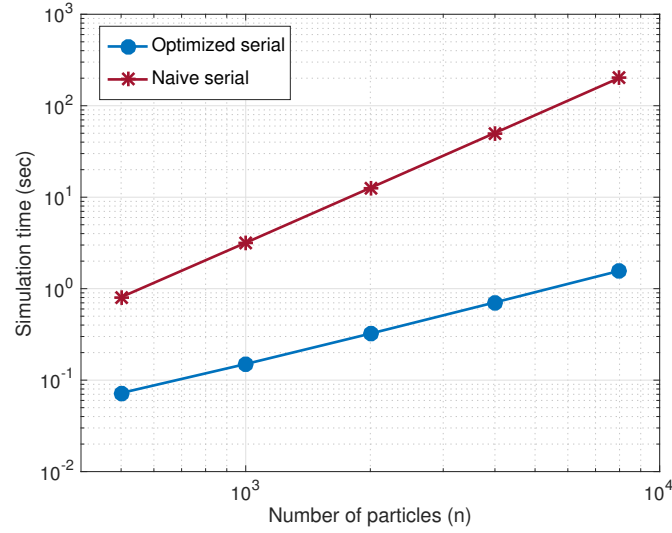


Figure 7.4: log-log plot of simulation time for different number of particles for the naive serial code and the modified one with binning method.

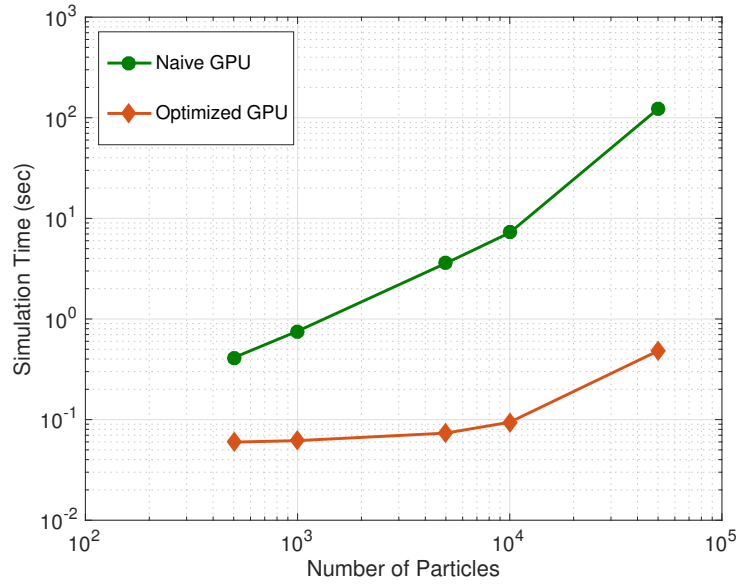


Figure 7.8: log-log plot of simulation time versus different number of particles for naive and optimized CUDA GPU

In Fig. 7.8 we have shown the performance of the optimized GPU code versus the naive CUDA code. As explained in section 7.2.8 partitioning the domain in parallel and assigning many threads to them efficiently speeds up the simulation.

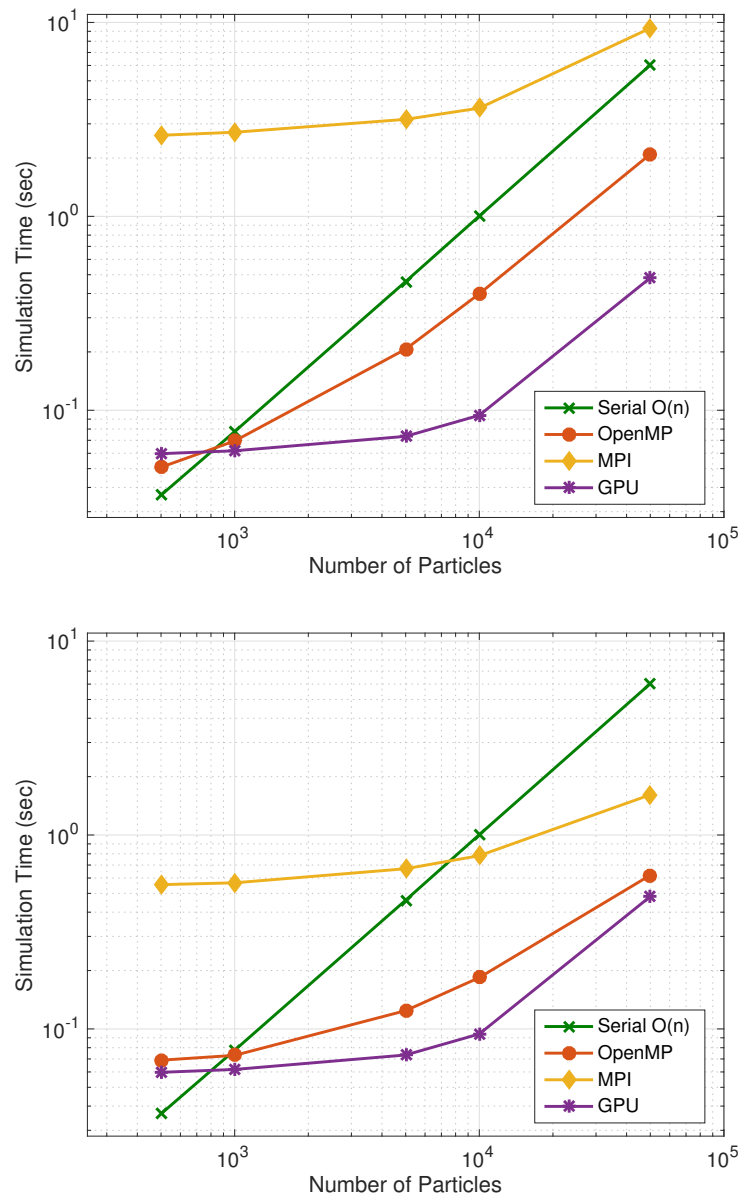


Figure 7.5: log-log plot of simulation time versus different number of particles for a fixed number of threads of 4 (top) and 24 (bottom)

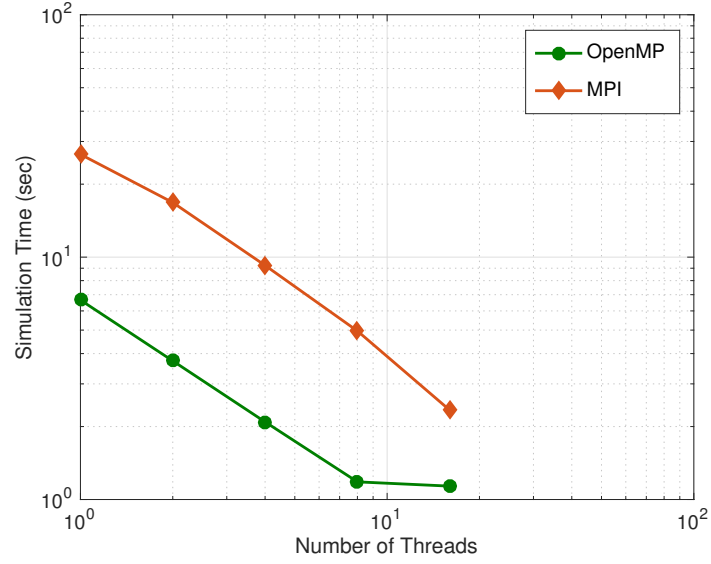


Figure 7.6: log-log plot of simulation time versus different number of threads for a fixed number of particles = 50000

### 7.3.2 Scaling efficiency

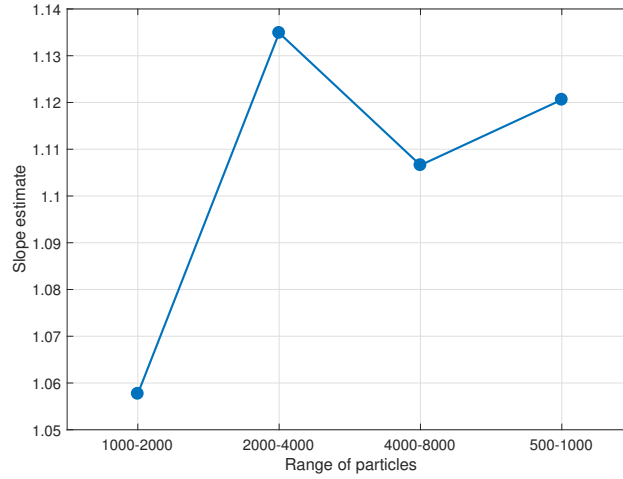


Figure 7.9: Slopes estimates for Algorithm 3 implementation

In the context of high performance computing there are two common notions of scalability: The first is strong scaling, which is defined as how the solution time varies with the number of processors for a fixed total problem size. The second is weak scaling, which is

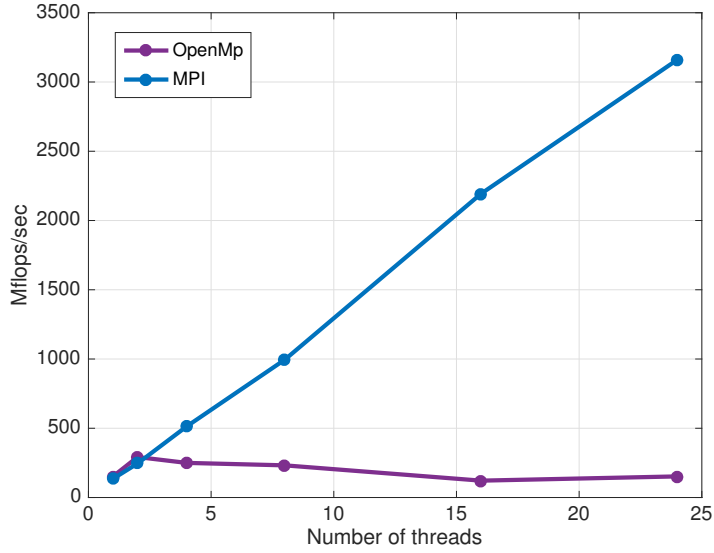


Figure 7.7: Comparison of Mflops/sec versus different number of threads for OpenMP and MPI

Table 7.2: Scaling factors for OpenMP algorithm

|                                   |       |
|-----------------------------------|-------|
| Average strong scaling speed up   | 4.78  |
| Average strong scaling efficiency | 0.66  |
| Average weak scaling efficiency   | 41.59 |

defined as how the solution time varies with the number of processors for a fixed problem size per processor.

Here we present our results regarding the scaling of our code for the cases of OpenMP and MPI. To evaluate the optimized serial algorithm, we consider the “slope estimate” of the line fit. In Fig. 7.9 we show the slopes estimates for different “ranges of number of Particles”. That is, we show the slope estimate when we are increasing the number of particles from 500 to 1000, from 1000 to 2000, and so on and so forth. This shows that our implementation is essentially of *linear* complexity.

In Fig. 7.10 results for average speed up, strong scaling, and weak scaling efficiency, respectively, are presented versus the number of threads/processors. The average of each factor across number of threads/processors are also tabulated in Tab. 7.2 for easier comparison.

Figure ?? shows the same comparison as above for MPI implementation. As we expected, the OpenMP implementation does not scale in a great way as the number of processors is increased, due to the “shared memory” overhead whereas MPI scales much better.

As discussed above, our MPI code partition the plane into vertical stripes and assign each

Table 7.3: Scaling factors for MPI algorithm

|                                   |       |
|-----------------------------------|-------|
| Average strong scaling speed up   | 7.67  |
| Average strong scaling efficiency | 0.8   |
| Average weak scaling efficiency   | 15.39 |

stripe to a processor. This, however, is not theoretically the most efficient way to partition the space. To minimize the number of communication, we want the total perimeter of the all the partitions to be minimized as the perimeter tells us, in expectation, how many particles need to be sent to more than one processors. For some number of processors, it is clear that there are better partitions; for example, if the number of processors is not a prime, then dividing the space both horizontally and vertically into grids yields a smaller perimeters. (Oudet, 2011) provides a partition that approximates the optimal perimeter. It is unclear, however, whether this is optimal and therefore, can be a further direction to investigate.

Moreover, as we stated above that our CUDA code just uses simple arrays in contrast to vectors used by our serial, OpenMP and MPI programs. In almost all instances, arrays are faster than vectors, as vector needs to dynamically allocate memories. Therefore, it can be explored to see what kind of performance improvement we can get if we replace vectors by arrays in our serial, OpenMP and MPI implementations.

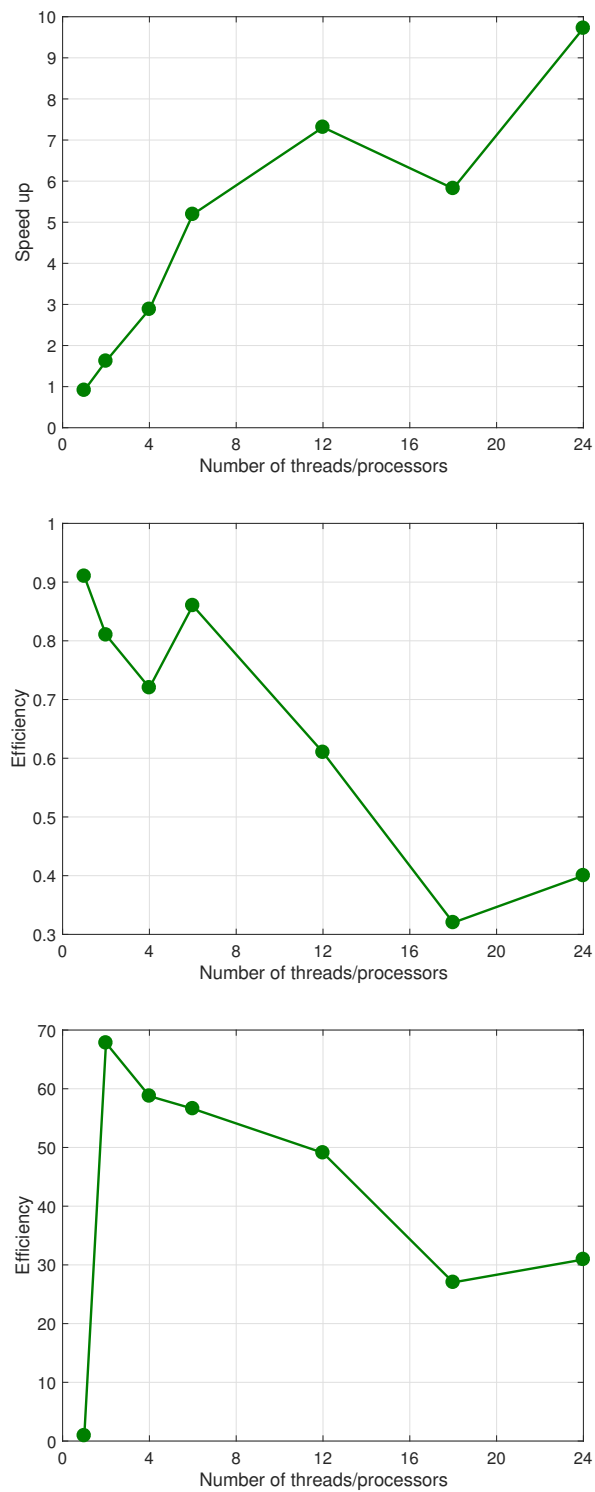


Figure 7.10: OpenMP scaling: from top image to below: strong scaling of speed up, strong scaling of efficiency, weak scaling of efficiency



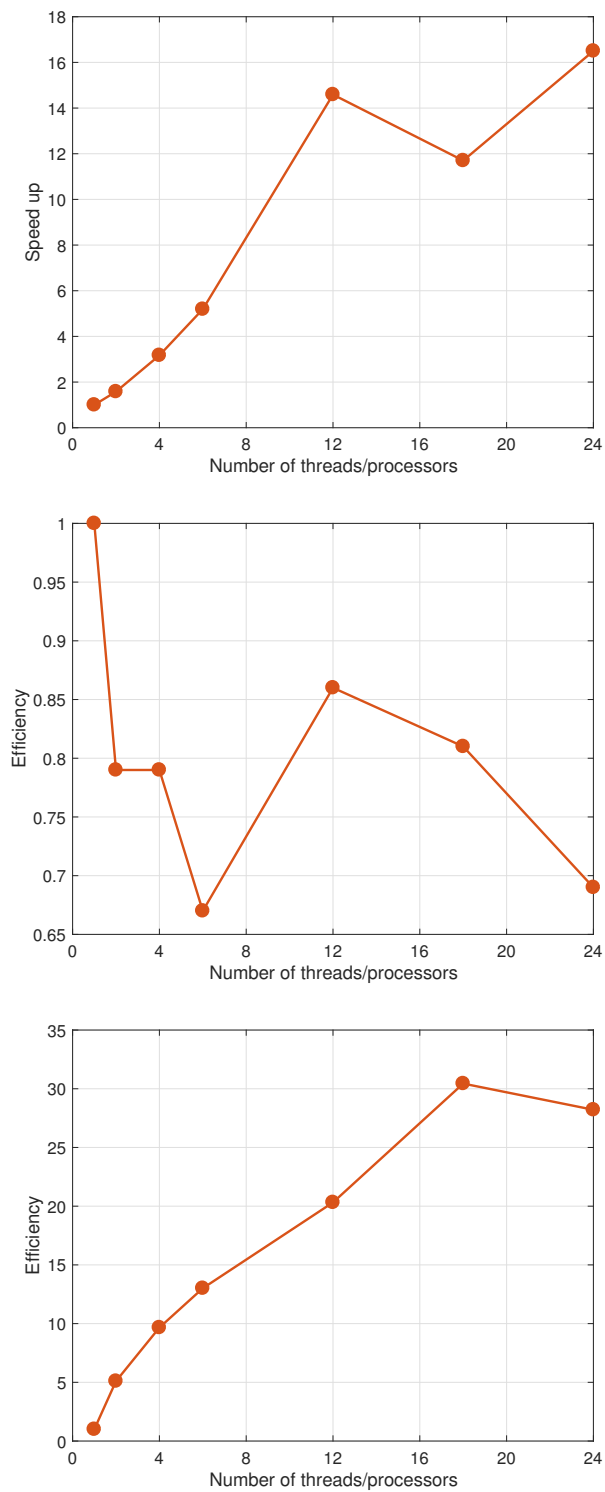


Figure 7.11: MPI scaling: from top image to below: strong scaling of speed up, strong scaling of efficiency, weak scaling of efficiency

## Part II

# *Learning* Mechanical Behavior of Materials

## Chapter 8

# Materials Modeling via Graph Neural Networks

In the first part of this thesis, we presented peridynamic formulation for materials and proposed how we can use PD theory to model mechanical behaviors such as crack propagation in ductile materials (5), and ballistic impact (6). In the second part of the thesis, we will explain how advances in deep learning can enhance studying materials' behavior and reduce computational cost. In this chapter, we briefly introduce the basics of neural networks, graph neural networks, and how we can combine peridynamic and neural networks to predict behavior of a wide range materials undergoing fracture.

### 8.1 Introduction to Neural Networks

Computational mechanics aims at finding approximate solutions for a variety of ordinary or partial differential equations which formulate various physical or chemical phenomena. For feasibility reasons, these approaches discretize the materials into their smallest units which can represent the the material behavior. However, there is a computational limit in the number of units a material can be broken into in real-world applications. Moreover, we simplify the non-linear equations by linearizing them and assuming perfect boundary or initial conditions to be able to derive approximate solutions. Nevertheless, these perfect assumptions and linearizations oversimplify the problems as well as causing excessive computational overhead. As a result, with the recent advances in computing power, data-driven and machine learning algorithms have been revisited and found to be extraordinarily successful in extracting correlations in high-dimensional parameter spaces. With the striking progress of graphical processing units (GPU), deep learning in particular won overwhelming victory over any other technique which has greatly impacted other fields in science and technology such as computer vision (Krizhevsky *et al.*, 2012; He *et al.*, 2016), medical and health-care applications (Lee *et al.*, 2017), autonomous driving (Ebrahimi *et al.*, 2017), generating high resolution images (Dong *et al.*, 2014; Azadi *et al.*, 2018), computational fluid mechanics

(Raissi *et al.*, 2020), astronomy (Chen *et al.*, 2014), agriculture (Kamilaris & Prenafeta-Boldú, 2018), business and finance applications (Heaton *et al.*, 2017), and computational solids mechanics (Oishi & Yagawa, 2017), etc. and is being widely used for collecting and analyzing enormous amount of data (Sinha *et al.*, 2019).

Machine learning has a long history (Samuel, 1959; Murphy, 2012) and many aspirations. It is composed of computer algorithms that develop automated mathematical models without being explicitly programmed about the underlying rules of the system. From small or large amount of data, machine learning can uncover complex patterns and use them to predict about outcomes of interest in future data. Machine learning has a lot in common with the fields of statistics and data mining but is different in the terminology, focus, and applications they have been used for (Murphy, 2012). Deep learning, is a specific kind of machine learning. In order to understand deep learning well, one must have a solid understanding of the basic principles of neural networks.

### 8.1.1 Feed-forward Neural Networks

Training nearly all deep learning algorithms follow a fairly simple recipe: collection of samples denoted as dataset, a *cost* function, a mapping function, and an optimization procedure to minimize the cost. Deep feed-forward networks, or multi-layer perceptrons (MLPs), are the key component of deep learning models. Let  $f$  be the mapping function parametrized by  $\theta$  that maps some input data  $\mathbf{x}$  to some output vector  $\mathbf{y}$  or a category of  $y$ , i.e.,  $\mathbf{y} = f(\mathbf{x}; \theta)$ .

Figure 8.1 shows a schematic of a neural network with an input and an output layer with scalar values and a *hidden layer* denoted as  $z$  with three number of *hidden units*.  $V$  and  $W$  are called *weight* matrices and denote the connection between two neighboring layers. No connections exist among units in the same layer as well as units in the non-neighboring layers. Each hidden layer is related to its precedent layer via a function called *activation* in the following form

$$\mathbf{z}(\mathbf{x}) = \phi(\mathbf{x}, \mathbf{V}) = g(\mathbf{V}(\mathbf{x})) \quad (8.1)$$

where  $g$  is a non-linear *activation* function and  $\phi$  is called the *hidden layer* which takes in  $\mathbf{x}$  and *weight* parameters (e.g.,  $V$  here).

From a probabilistic point of view a standard *fully-connected* (FC) neural network can be defined as a probability function  $p(y|\mathbf{x}, \theta)$ . To compute  $p$ , we build a network parametrized by  $\theta$  and try to find the best set of weight parameters such that an objective function is satisfied or in other words, a cost or error function is minimized. For that, we initialize our network using a generic Gaussian distribution with some known mean and variance:

$$p(y|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{y}|\mathbf{W}^T \phi(\mathbf{x}, \mathbf{V}), \sigma^2 \mathbf{I}) \quad (8.2)$$

We can think of  $\theta$  as a set of features describing the dataset. The dominant approach in all scientific fields has been always trying to use laws from physics to explain the *pattern* of an event. But deep learning provides an alternative way of automatically *learning* this

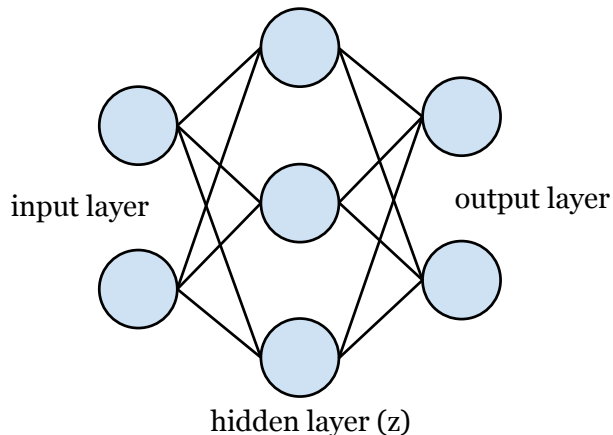


Figure 8.1: A multi-layer perceptron (MLP) with a single hidden layer

mapping function by defining it as an optimization problem which computes the posterior using gradient-based methods.

Neural nets are called *networks* because each network is composed of a chain of layers, each of them mapping an input to an output which serves as an input for the next layer. Input data flow through a neural network via connections between units, starting from the input layer, and then through hidden layers, finally to the output layer, and the last layer outputs the processed data as a desired outcome which can be a real-valued number or an integer defining a category the output belongs to. The former case is called a *regression* and the latter is a *classification* problem. Equation 8.2 represent the parameters of a regression problem.

## 8.2 Graph Representation of a Material

Graph Networks (GN) are powerful in representing arbitrary (pairwise) relational structures. Graph Neural Networks (GNN) are a class of functions for relational reasoning over graph-structured representations and have been widely investigated in the past decade (Gori *et al.*, 2005; Scarselli *et al.*, 2005; Scarselli *et al.*, 2008a; Li *et al.*, 2015). They have been successfully applied in understanding tasks with rich relational structure such as robotics (Sukhbaatar, Fergus, *et al.*, 2016; Kipf *et al.*, 2018), dynamics of systems (Sanchez-Gonzalez *et al.*, 2018; Battaglia *et al.*, 2016; Van Steenkiste *et al.*, 2018), quantum chemistry (Gilmer *et al.*, 2017), fracture mechanics (Schwarzer *et al.*, 2019), and materials science (Xie & Grossman, 2018). For a complete survey on early works on graph neural networks readers are referred to (Scarselli *et al.*, 2008b) whereas (Bronstein *et al.*, 2017) provides a comprehensive review on geometric deep learning approaches.

As discussed in chapter 2, we used the equation of motion defined in Eq. 2.30 to solve for displacement of particles over a course of several time-steps. The key component of Eq.

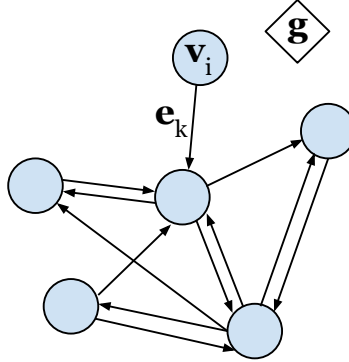


Figure 8.2: A graph sample with  $\mathbf{v}_i$  as a node,  $\mathbf{e}_k$  as an edge, and  $\mathbf{g}$  as the global attributes such as dynamic force which can be encoded as a vector or even another graph.

2.30 was the peridynamic force which was derived through strain energy density function shown in Eq. (2.21). However, we now intend to learn all the dynamic interactions using the data collected in the PD experiments in the previous chapters. We argue that a discretized peridynamic material can be represented with a bidirected graph where nodes correspond to the particles and edges are the peridynamic bond between two particles belonging to the same *family*.

The main module of GN framework is the GN block, referred to as *graph2graph*, which takes a graph as input, and outputs another graph after performing some computations over its structure. Figure 8.2 depicts a graph defined as  $G = (\mathbf{g}, V, E)$  composed of nodes denoted as  $\mathbf{v}_i$ , edges as  $\mathbf{e}_k$ , and the global attributes as  $\mathbf{g}$ .

The  $\mathbf{g}$  is a global attribute; for example,  $\mathbf{g}$  can contain constant material properties. The  $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$  is the set of nodes (vertices), where each  $\mathbf{v}_i$  with  $i = \{1, \dots, N^v\}$  represents properties of node  $i$  such as position, velocity, mass, etc. Lastly,  $E = \{\mathbf{e}_k, s_k, r_k\}_{k=1 \dots N^e}$  is the set of  $N^e$  number of edges where each  $\mathbf{e}_k$  represents attributes of the edge. For each  $\mathbf{e}_k$ , there exists a sender and receiver node indexed as  $s_k$  and  $r_k$ , respectively. In this work, each sender is also a receiver in an edge because two interacting material points exert non-equal force on each other in opposite directions due to Newton's third law.

Every GN has an *update* mechanism composed of three functions to update properties of nodes, edges, and global features denoted as  $\phi^v$ ,  $\phi^e$ , and  $\phi^g$ , respectively. There exists three *aggregation* functions,  $\rho$  functions which are all defined below:

$$\begin{aligned} \mathbf{e}'_k &= \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{g}) & \bar{\mathbf{e}}'_i &= \rho^{e \rightarrow v}(E'_i) \\ \mathbf{v}'_i &= \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{g}) & \bar{\mathbf{e}}' &= \rho^{e \rightarrow g}(E') \\ \mathbf{g}' &= \phi^g(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{g}) & \bar{\mathbf{v}}' &= \rho^{v \rightarrow g}(V') \end{aligned} \quad (8.3)$$

Each  $\rho$  function reduces a set of input to aggregated information represented with a single element. Conventionally  $\rho$  can be an element-wise operator performing summation, mean, or maximum.

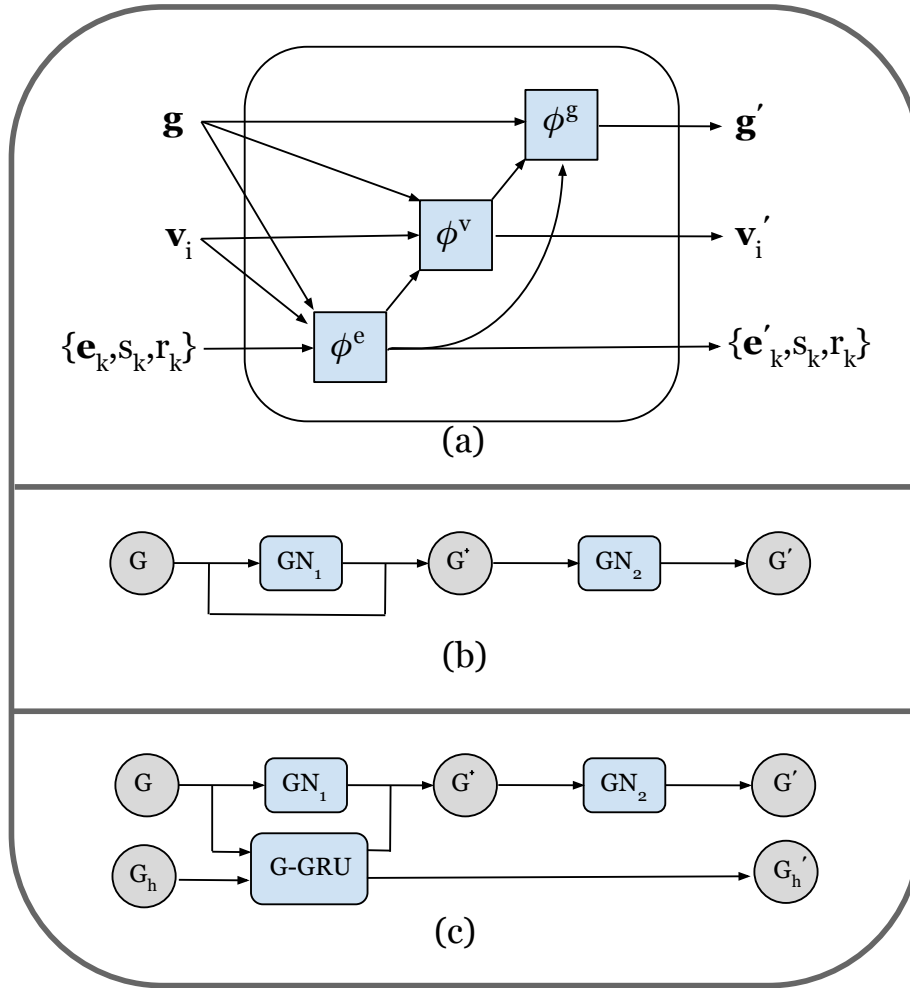


Figure 8.3: (a) A *graph2graph* network takes a graph as input and returns a graph with the same structure but different edge, node, and global features as output. (c) A feed-forward GN-based forward model for learning one-step predictions. (d) A recurrent GN-based forward model.

We also define two graphs that we intent to use to model static and dynamic properties of our peridynamic system:

**Static graph ( $G_s$ ):** as the name suggests, static or constant information about the particles including material properties such as Bulk modulus, Shear modulus, and simulation time-step, critical stretch bond.

**Dynamic graph ( $G_d$ ):** this graph contains information about the position ( $\mathbf{x}, \mathbf{y}$ ), velocity of the particles at each time step ( $\mathbf{v}_x, \mathbf{v}_y$ ), bond stretch ( $s$ ), and the peridynamic force ( $\mathbf{f}_x, \mathbf{f}_y$ ) between two particles.

Figure 8.3a shows the core GN architecture used in this work as *graph2graph* modules

**Algorithm 7** GRAPH NEURAL NETWORKS

---

```

1: Input:  $G = (\mathbf{g}, \{\mathbf{v}_i\}_{i=1\dots N^v}, \{\mathbf{e}_k, s_k, r_k\}_{k=1\dots N^e})$ 
2: for each edge  $\{\mathbf{e}_k, s_k, r_k\}_{k=1\dots N^e}$  do
3:   Collect sender and receiver nodes  $\mathbf{v}_{s_k}$  and  $\mathbf{v}_{r_k}$ 
4:    $\mathbf{e}'_k \leftarrow \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{g})$  ▷ Output edges
5: for each node  $\mathbf{v}_i$  do
6:   Aggregate  $\mathbf{e}'_k$  per receiver
7:    $\mathbf{v}'_i \leftarrow \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{g})$  ▷ Node-wise features
8: Aggregate all edges and nodes
9:  $\mathbf{g}' \leftarrow \phi^g(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{g})$  ▷ Global features
10: Output:  $G' = (\mathbf{g}', \{\mathbf{v}'_i\}_{i=1\dots N^v}, \{\mathbf{e}'_k, s_k, r_k\}_{k=1\dots N^e})$ 

```

---

with three sub-modules denoted as  $\phi^e$ ,  $\phi^v$ , and  $\phi^g$  each can be in form of MLP or recurrent neural network (RNN) configurations. Fig. 8.3b shows a single feedforward GN pass which can be viewed as one step of message-passing on a graph (Gilmer *et al.*, 2017), where  $\phi^e$  is first applied to update all edges,  $\phi^v$  is then applied to update all nodes, and  $\phi^g$  is finally applied to update the global feature. See Algorithm 7 for details.

The reason we used two GNs (see Fig. 8.3b) was to allow all nodes and edges to communicate with each other through the  $\mathbf{g}$  output from the first GN. Similar to (Sanchez-Gonzalez *et al.*, 2018), we found performance advantages over using single GN only. We also used a recurrent GN-based inference model, which observes the dynamic states of a trajectory. It takes a sequence of dynamic state graphs ( $G_d$ ) and after  $T$  timesteps returns another graph denoted as  $G'(T)$  which is then concatenated with  $G_d$  and fed to the forward model. Figure 8.3c shows the RNN core takes in the concatenation of  $G_d$  and  $G_h$  and returns  $G'$  as well as the updated hidden graph denoted as  $G'_h$ . The full architecture can be trained jointly end-to-end and used for making prediction on behavior of materials never seen before based on a set of behavior *history* observed by other materials. The experimental setup and training procedures are described in the following section.

### 8.3 Experimental Setup

In this section we explain our approach to use our PD models derived in chapter 5 and chapter 6 to predict dynamic crack propagation accurately in unknown materials without the need to run any computational or physical experiment on them.

Our training data is composed of peridynamic results for 60 isotropic materials (ductile and brittle) undergoing two dimensional fracture due to tensile loading of different magnitudes with or without a pre-crack or a pre-existing hole placed randomly on the plate. We took material properties for 60 real-world materials from MatMatch website (Material Match 2020), 50 of which we used for training data and 10 for test. We considered a wide range of materials including ceramics, metals, and polymers. Dataset collecting for each



material, and hence each peridynamic simulation, contains a sequence per particle (graph node) position, velocity, damage index, yield index (ductile materials only), and dynamic force as well as edge-based information such as bond stretch. Our goal is to generate a time series of graphs representing the evolution of fracture or any other discontinuity propagation in an unseen material by imitating the peridynamic model used to model fracture in the training data and capturing complex non-linear relationship in the structure. Our network architecture utilizes the temporal aspect of RNNs and their capability in learning a *memory* for constitutive occurrences. Once being trained, our model can take an initial set up for the material at time  $t_0$ , for instance the time-step, material properties such as Shear and Bulk modules, initial coordinates and velocity of the particles, and return the sequence for location, velocity, dynamic force, and damage index of each particle over a course of several timesteps. As with the peridynamic models, once a bond between two nodes reaches its critical stretch limit, it breaks and never recovers. This phenomenon guides the crack propagation and is tracked using the damage index for each node, ranging from 0 (intact bond) to 1 (broken bond), denoting the ratio of the broken to total number of bonds. For ductile material we also keep track of yield index ( $\lambda$ ), defined in 5.6 per bonds where  $\lambda = 0$  indicates an unyielded and  $\lambda = 1$  represents a fully yielded or plastically deformed bond.

Our architecture includes  $N^v$  graphs, each graph representing a *family* of neighboring particles connected through an adjacency matrix. The output is also a series of the same length graphs which are shifted over by one timestep. We have used the followings for architecture in each sub-module:

- Edge MLP: 2 or 3 hidden layers. 256 to 512 hidden cells per layer.
- Node and Global MLP: 2 hidden layers. 128 to 256 hidden neurons per layer.
- Updated edge, node and global size: 128
- (Recurrent models) Node, global and edge size for state graph: 20

We used a standard L2-norm loss between the predictions and expected values (ground truth data) for position and linear velocity. For optimizer we used Adam (Kingma & Ba, 2014) with batch size of 128 graphs/graph sequences on a single GPU.

## 8.4 Results

We have been able to train our model to predict the trajectory sequence with  $0.4 \pm 0.01\%$  MSE error for the training data and  $1.3 \pm 0.02\%$  for test set. Qualitative results also provide a good insight on how the predictor behavior matches between peridynamic and GNNs. Below we have provided snapshots of the predicted displacement, damage index, and yield index (5.6) (for ductile materials) for both success and failure cases.

Figure 8.4 shows a comparison between results from PD modeling serving as the ground truth against predictions made by our trained GNN. Figure 8.4a shows the plate at  $t_0$  with no

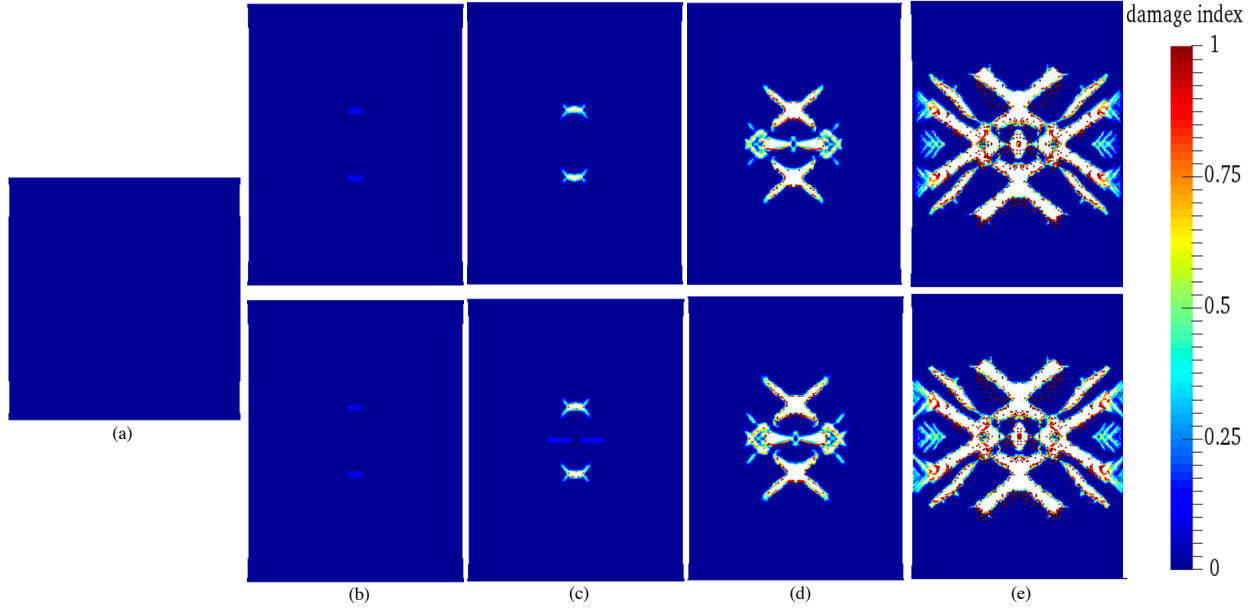


Figure 8.4: Damage index per particle for a plate with no pre-existing defect. (a) shows the initial plate at time  $t = 0$ . (b-e) damage at timesteps  $t = 5\Delta t$ ,  $t = 15\Delta t$ ,  $t = 20\Delta t$ ,  $t = 25\Delta t$ , respectively

pre-existing discontinuity of any kind. In columns (b), (c), (d), and (e), we show a sequence of heatmaps at different time-steps. Top row visualizes ground truth data for damage index per particle whereas bottom row shows those values predicted by our trained GNN. It should be noted that model has neither seen a ductile material of the same kind during training nor a ductile material without any pre-existing defect. GNN predictions seem to agree well with PD results of this material showing a significant generalization capability to loading and geometry conditions never seen before.

Figure 8.5 depicts another example where we have compared damage growth produced by PD and GNN predictions. Similar to the previous case, this setup did not exist in the training data but GNN is able to generalize well in predicting where damage begins and where it leads to.

Figure 8.6 shows same scenario but visualizes results for yield index. GNN model and ground truth results here do not align with each other in column (b) where GNN model predicts larger values for yield index for particles on the path along the center-line. Wrong predictions for yield index at timestep  $t = 5\Delta t$  is followed by more pronounced yield in the following timesteps where the error propagates and ultimately deviate by large amounts from the reference values. This effect is also visualized in Figure 8.7 where the disagreement starts at crack initiation time leading to predicting drastically larger values of plastic deformation at crack tips and ultimate faster rupture of the entire plate.

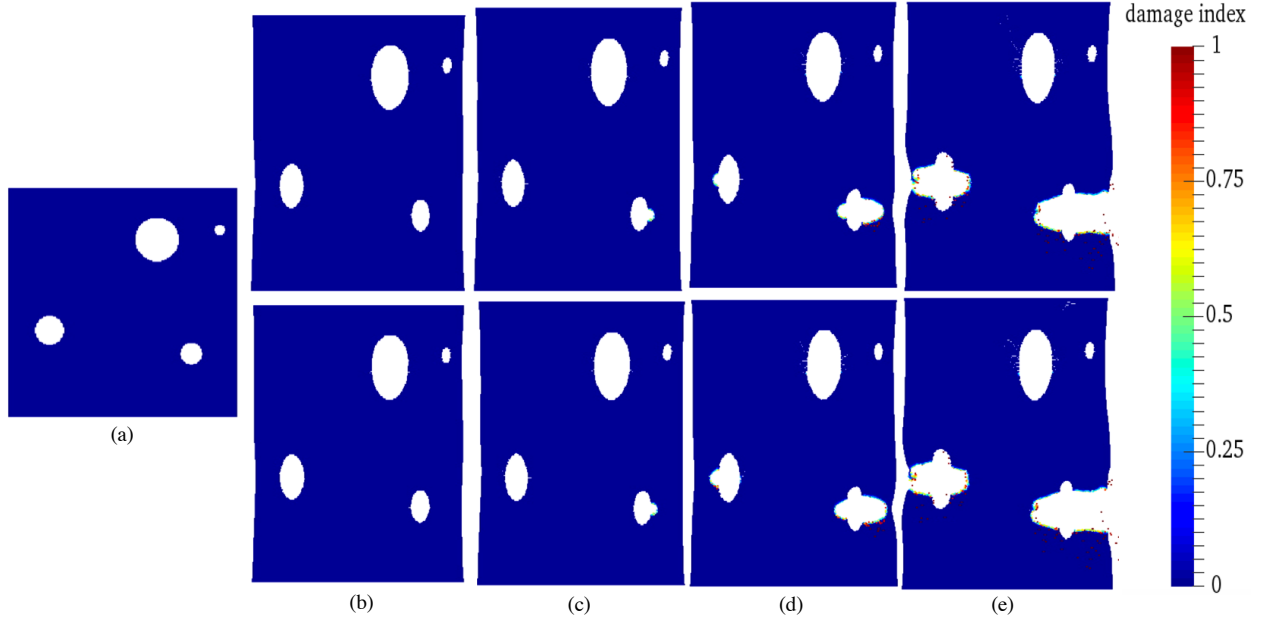


Figure 8.5: Damage index per particle for a plate with randomly placed pre-existing holes of different size. (a) shows the initial plate at time  $t = 0$ . (b-e) damage at timesteps  $t = 5\Delta$ ,  $t = 15\Delta$ ,  $t = 20\Delta t$ ,  $t = 25\Delta t$ , respectively

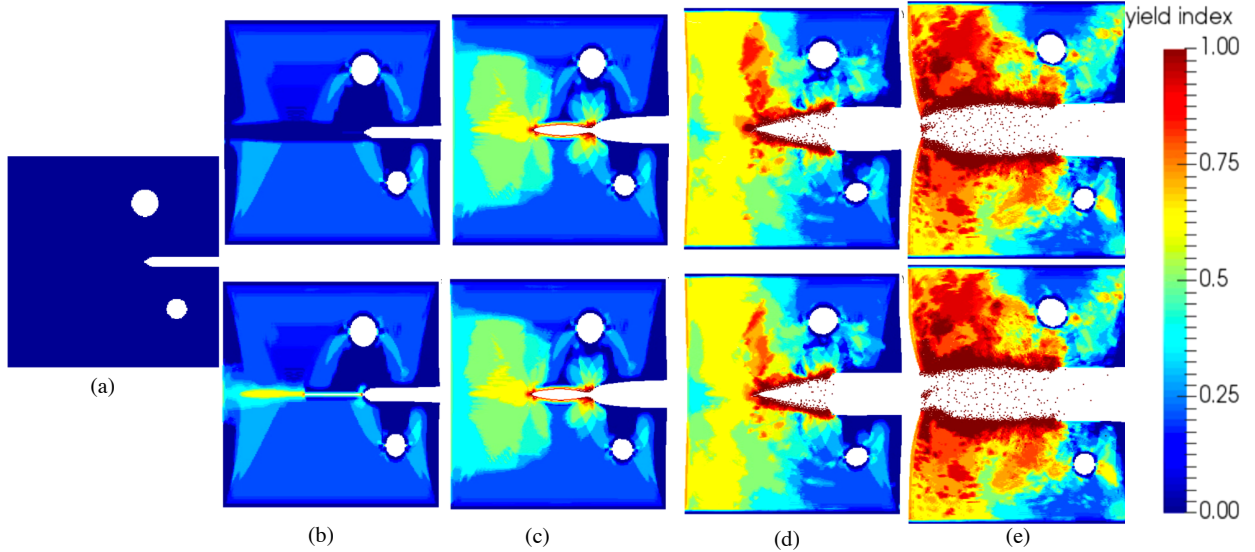


Figure 8.6: Yield index per particle for a plate with randomly placed pre-existing holes of different size. (a) shows the initial plate at time  $t = 0$ . (b-e) yield effect at timesteps  $t = 5\Delta$ ,  $t = 15\Delta$ ,  $t = 20\Delta t$ ,  $t = 25\Delta t$ , respectively

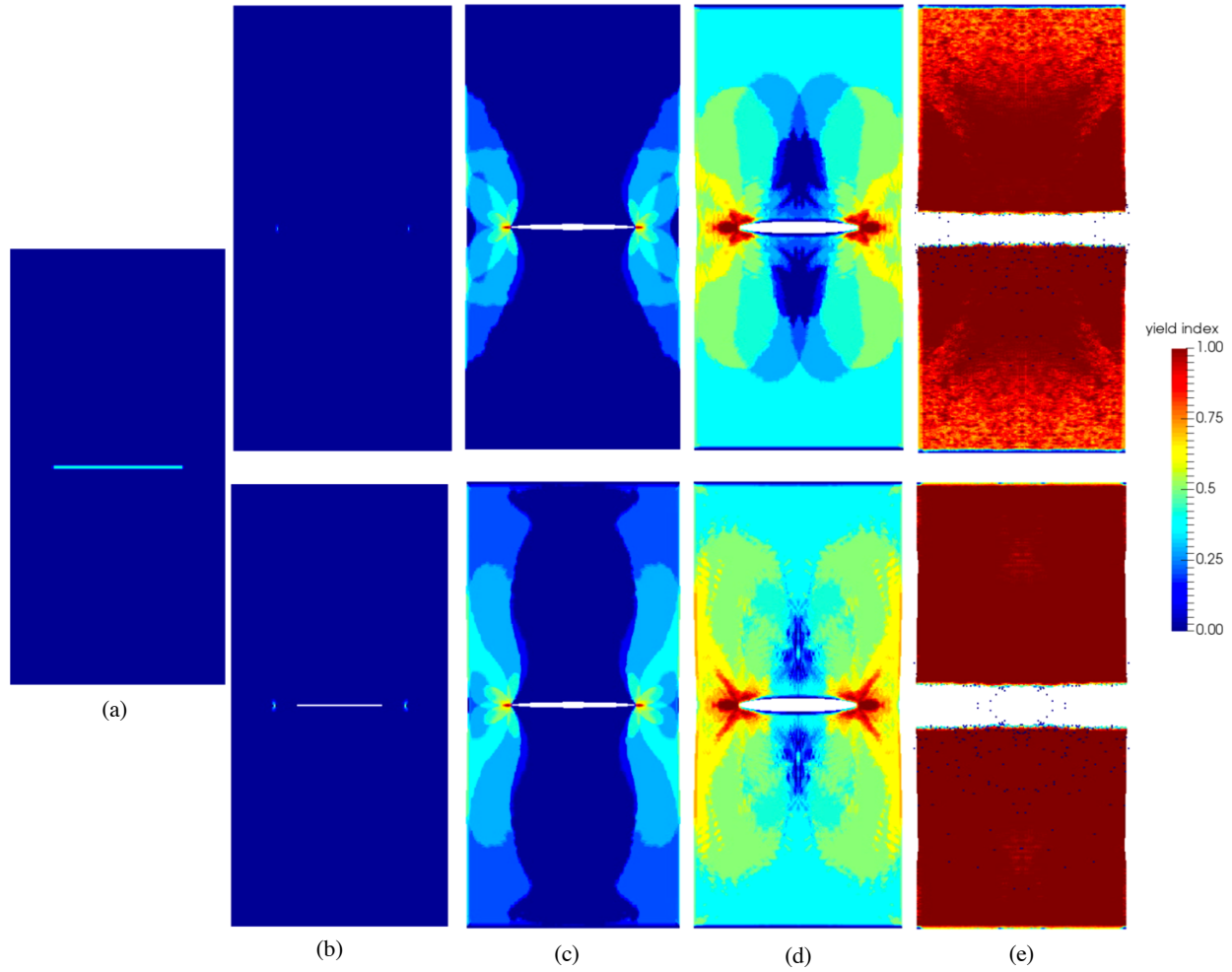


Figure 8.7: Yield index per particle for pre-cracked plate. (a) shows the initial plate at time  $t = 0$ . (b-e) yield effect at timesteps  $t = 5\Delta$ ,  $t = 15\Delta$ ,  $t = 20\Delta$ ,  $t = 25\Delta$ , respectively

## Chapter 9

# Conclusion and Future Work

### 9.1 Discussion of Contributions

This dissertation aimed to explore two seemingly different paths for modeling mechanical behavior of materials: 1) Peridynamic (PD) theory as a relatively new alternative framework to classical continuum theory and 2) *deep learning* based approaches using graph neural networks. In the first part, we mainly focused on formulating PD in physical problems which had been previously analyzed either with numerical approaches such as FEM, or with experimental observations. In the second part, we developed a *learnable* model to predict materials' behavior in order to bridge the gap between statistical representation learning and physics based approaches in modeling worlds events. Summary of our key contributions is provided as follow:

#### 9.1.1 Part I: Peridynamic Analysis of Materials Behavior

Our contributions in part I of this dissertation are summarized as below:

- Chapter 3: we developed PD models to study wear and friction in ultra thin films at nanoscale. Prior continuum-based approaches were not able to account for local differences in nanostructure whereas MD simulations were limited by high computational cost, model size, and choice of potential functions used to describe atomic interactions. We showed, for the first time, that PD can successfully model sliding friction and wear process of amorphous films thinner than 100nm thickness (Ebrahimi *et al.*, 2015a). We were able to compute coefficient of friction that matched well with the experimental results under identical loading conditions. The closest PD work in this line of research was (Celik *et al.*, 2009) in which bond-based PD was used to find material properties of polymeric films deposited on soft substrates where no damage was allowed and comparisons were made with MD simulations. Our work was different from (Celik *et al.*, 2009) in using state-based PD, modeling friction, as well as modeling nano-mechanical wear depth caused by hard indentation of diamond tips.

- Chapter 4: we analyzed three modes of delamination in compressed elastic thin films on compliant substrates: wrinkling, buckling and concomitant wrinkling and buckle-delamination (Ebrahimi *et al.*, 2015b). The closest work that explored delamination using PD was (Hu *et al.*, 2015) in a parallel study where the goal was to derive critical stretch bond based upon the strain energy release rate due to the increase in interlaminar tension and shear at the contact surface. In our study, we used strain energy and its gradient to determine the key roles in emergence of wrinkling and the transition to concomitant wrinkling and buckling, and eventually to buckling. We derived *phase* diagrams that determines the delamination mode and hence the change in mechanical behavior across different properties mismatch for films undergoing compression on top of compliant substrates. Our PD results correlate well with the analytical and experimental results available in the literature.
- Chapter 5: one of our major contributions in this dissertation was developing a new ordinary stated-based peridynamic formulation for plasticity using mechanical sub-layer method originally proposed by (Besseling, 1953) and extended by (Zienkiewicz *et al.*, 1973) and (Pian, 1987). Our plasticity model differs from the previous work by (Taylor, 2008) in defining yield criterion and using more general form of state-based theory. We introduced *yield index* to measure the ductile behavior of particle before their bond breaks. We predict ductile fracture for center-cracked plate and compact tension problems in 2D where our PD results agreed well with computational and experimental work by (Areias *et al.*, 2014) and (Samal *et al.*, 2009), respectively. Moreover, our PD results for J-integral calculation are compared with those from benchmark problems and an excellent agreement is achieved.
- Chapter 6: we used PD in a real-world practical application to design body armor systems. PD has been proven successful in modeling dynamic fracture and impact analysis. We took this line of research one step further by creating two PD models for 1) rigid and 2) non-rigid bullets. In the former, we used a bio-inspired structure by Abalone, a marine snail that protects itself with its convex she with incredibly stiff layered structure. In the former case, non-rigid bullet, we adopted our PD plasticity model derived in 5 and modeled the backing panel behavior in maximizing the energy absorption while being cost and weight efficient. We analyzed the composite structure of the backing panel by examining different unidirectional and multidirectional fiber reinforced composites. Our analysis was specific to the type of projectile for which we designed the armor, such as handgun bullets, rifle bullets, and heavy bullets and we were able to analyze most popular armor systems and proposing using new materials for the front and backing layers.
- Chapter 7: in the last chapter of part I, we parallelized PD algorithm on CPU using shared memory (OpenMP) and distributed memory (MPI) as well as on graphics processing unit (GPU) via NVIDIA CUDA architecture. We performed extensive experiments to determine the scaling factor of each parallelization technique and found

the bottlenecks in designing parallel PD models and the best parallization practice for them. We used C++ here as it could offer the most efficient object-oriented setting for this study.

### 9.1.2 Part II: *Learning* Mechanical Behavior of Materials

Our contribution in part II is described below:

- Chapter 8: The availability of computing power and emergence of the new machine-learning based approaches in the recent years motivated us to revisit the problems investigated in part I from a data-driven perspective using deep learning. We proposed to use graph neural networks (GNNs) to develop a framework to directly learn material behavior under complex loading conditions using the data provided by Peridynamic simulations providing a universal, generalizable, and interpretable representation of solid materials including ceramics, metals, and polymers. The interpretability is due to the excellent capability of GNNs in extracting the contributions of graph edges, i.e., PD particles, to global properties.

## 9.2 Future Perspectives

Now, we describe some of the future directions that immediately follow from the quantitative and qualitative analysis provided in this work:

- **Directions to expand PD theory:** this research clearly illustrates benefits of using a non-local continuum-based approach as an alternative to the classical theory, but it also raises the question of how PD formulation has to change should any of the assumptions we made about *ordinary state-based* PD do not hold true anymore. Our major drawback in using *non-ordinary* state based PD was the lack of results from controlled experiments in the literature to verify PD simulations with. One can investigate this path and design and conduct experiments that can be used to verify *non-ordinary* state based PD which covers modeling a broader range of materials and their complex interactions.
- **Directions to expand learning mechanical behavior of solids materials:** This work introduced a new class of learnable forward and inference models, based on graph neural networks (GNN), which implements the relational behavior of material points. Across a range of experiments we found that these models are surprisingly accurate, robust, and generalizable when used for prediction, in challenging unseen loading conditions. While our GNN-based models were most effective in systems with common discontinuities (e.g., cracks, holes), they were less successful at the presence of abnormal defects which were never seen during training. Our approach also does not address a common problem of errors compound over long trajectory predictions. Some

key future directions include using our approach for material identification in real-world settings, supporting experimenting in hazardous environments. This work takes a key step towards realizing the promise of machine learning based methods by exploiting compositional representation learning within a powerful statistical learning framework, and opens new paths for robust, efficient, and general-purpose patterns of reasoning and decision-making.

In the words of Alan Turing,

*We can only see a short distance ahead,  
but we can see plenty there that needs to be done.*



# Bibliography

1. Agwai, A., Guven, I. & Madenci, E. Crack propagation in multilayer thin-film structures of electronic packages using the peridynamic theory. *Microelectronics Reliability* **51**, 2298–2305 (2011).
2. Agwai, A., Guven, I. & Madenci, E. *Damage prediction for electronic package drop test using finite element method and peridynamic theory* in *59th Electronic Components and Technology Conference, IEEE* (2009), 565–569.
3. Agwai, A., Guven, I. & Madenci, E. Drop-shock failure prediction in electronic packages by using peridynamic theory. *IEEE Transactions on Components, Packaging and Manufacturing Technology* **2**, 439–447 (2012).
4. Agwai, A., Guven, I. & Madenci, E. *Peridynamic theory for failure prediction in multilayer thin-film structures of electronic packages* in *58th Electronic Components and Technology Conference, IEEE* (2008), 1614–1619.
5. Alali, B. & Lipton, R. Multiscale dynamics of heterogeneous media in the peridynamic formulation. *Journal of Elasticity* **106**, 71–103 (2012).
6. Alder, B. J. & Wainwright, T. Studies in molecular dynamics. I. General method. *Journal of Chemical Physics* **31**, 459–466 (1959).
7. Aliabadi, M. H. & Rooke, D. The boundary element method. *Numerical Fracture Mechanics*, 90–139 (1991).
8. Areias, P., Rabczuk, T. & Camanho, P. Finite strain fracture of 2D problems with injected anisotropic softening elements. *Theoretical and Applied Fracture Mechanics* **72**, 50–63 (2014).
9. Askari, E. *et al.* Peridynamics for multiscale materials modeling. *Journal of Physics: Conference Series* **125**, 012078 (2008).
10. Askari, E., Xu, J. & Silling, S. *Peridynamic analysis of damage and failure in composites* in *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006).
11. Azadi, S., Pathak, D., Ebrahimi, S. & Darrell, T. Compositional gan: Learning conditional image composition (2018).
12. Banks-Sills, L. Application of the finite element method to linear elastic fracture mechanics. *Appl. Mech. Rev* **44**, 447–461 (1991).

13. Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., *et al.* *Interaction networks for learning about objects, relations and physics* in *Advances in neural information processing systems* (2016), 4502–4510.
14. Bazhenov, S. "Dissipation of energy by bulletproof aramid fabric". *Journal of materials science* **32**, 4167–4173 ((1997)).
15. Belytschko, T. & Black, T. Elastic crack growth in finite elements with minimal remeshing. *International journal for numerical methods in engineering* **45**, 601–620 (1999).
16. Besseling, J. *A Theory of Plastic Flow for Anisotropic Hardening in Plastic Deformation of an Initially Isotropic Material* (Nationaal Luchtvaartlaboratorium, 1953).
17. Bobaru, F. Influence of van der Waals forces on increasing the strength and toughness in dynamic fracture of nanofibre networks: a peridynamic approach. *Modelling and Simulation in Materials Science and Engineering* **15**, 397–417 (2007).
18. Bobaru, F., Foster, J. T., Geubelle, P. H. & Silling, S. A. *Handbook of peridynamic modeling* (CRC Press, 2016).
19. Bobaru, F. & Hu, W. The meaning, selection, and use of the peridynamic horizon and its relation to crack branching in brittle materials. *International Journal of Fracture* **176**, 215–222 (2012).
20. Bobaru, F. & Silling, S. A. *Peridynamic 3D models of nanofiber networks and carbon nanotube-reinforced composites* in *Materials Processing and Design: Modeling, Simulation and Applications, NUMIFORM 2004, Proceedings of the 8th International Conference on Numerical Methods in Industrial Forming Processes* **712** (2004), 1565–1570.
21. Bobaru, F., Silling, S. A. & Jiang, H. *Peridynamic fracture and damage modeling of membranes and nanofiber networks* in *XI Int. Conf. Fract., Turin, Italy* (2005).
22. Bobaru, F. *et al.* Convergence, adaptive refinement, and scaling in 1D peridynamics. *International Journal for Numerical Methods in Engineering* **77**, 852–877 (2009).
23. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. & Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**, 18–42 (2017).
24. Burnett, P. J. & Rickerby, D. The mechanical properties of wear-resistant coatings: I: Modelling of hardness behaviour. *Thin Solid Films* **148**, 41–50 (1987).
25. Casiraghi, C., Robertson, J. & Ferrari, A. C. Diamond-like carbon for data and beer storage. *Materials Today* **10**, 44–53 (2007).
26. Cauchy, A. L. B. *Recherches sur l'équilibre et le mouvement intérieur des corps solides ou fluides, élastiques ou non élastiques* (1822).
27. Celik, E., Oterkus, E., Guven, I. & Madenci, E. *Mechanical characterization of ultra-thin films by combining AFM nanoindentation tests and peridynamic simulations* in *59th Electronic Components and Technology Conference, IEEE* (2009), 262–268.

28. Chan, E. P., Smith, E. J., Hayward, R. C. & Crosby, A. J. Surface wrinkles for smart adhesion. *Advanced Materials* **20**, 711–716 (2008).
29. Charitidis, C. Nanomechanical and nanotribological properties of carbon-based thin films: a review. *International Journal of Refractory Metals and Hard Materials* **28**, 51–70 (2010).
30. Chen, X. & Hutchinson, J. W. Herringbone buckling patterns of compressed thin films on compliant substrates. *Journal of applied mechanics* **71**, 597–603 (2004).
31. Chen, Y., Lin, Z., Zhao, X., Wang, G. & Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing* **7**, 2094–2107 (2014).
32. Dong, C., Loy, C. C., He, K. & Tang, X. *Learning a deep convolutional network for image super-resolution in European conference on computer vision* (2014), 184–199.
33. Dugdale, D. S. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids* **8**, 100–104 (1960).
34. Ebrahimi, S. & Phan, A.-V. Dynamic analysis of cracks using the SGBEM for elastodynamics in the Laplace-space frequency domain. *Engineering Analysis with Boundary Elements* **37**, 1378–1391 (2013).
35. Ebrahimi, S. & Phan, A.-V. Dynamic crack growth modeling technique based upon the SGBEM in the Laplace domain. *Acta Mechanica* **226**, 769–781 (2015).
36. Ebrahimi, S., Rohrbach, A. & Darrell, T. Gradient-free policy architecture search and adaptation. *arXiv preprint arXiv:1710.05958* (2017).
37. Ebrahimi, S., Steigmann, D. & Komvopoulos, K. Peridynamics analysis of the nanoscale friction and wear properties of amorphous carbon thin films. *Journal of Mechanics of Materials and Structures* **10**, 559–572 (2015).
38. Ebrahimi, S., Steigmann, D. & Komvopoulos, K. *Peridynamics Analysis of Wrinkling and Buckle-Delamination of Compressed Elastic Thin Films on Compliant Substrates in 13th US National Congress on Computational Mechanics* (2015).
39. Eringen, A. C. Linear theory of nonlocal elasticity and dispersion of plane waves. *International Journal of Engineering Science* **10**, 425–435 (1972).
40. Eringen, A. C. Nonlocal polar elastic continua. *International journal of engineering science* **10**, 1–16 (1972).
41. Eringen, A. C. & Kim, B. S. Stress concentration at the tip of crack. *Mechanics Research Communications* **1**, 233–237 (1974).
42. Eringen, A. C., Speziale, C. & Kim, B. Crack-tip problem in non-local elasticity. *Journal of the Mechanics and Physics of Solids* **25**, 339–355 (1977).
43. Foster, J. T., Silling, S. A. & Chen, W. An energy based failure criterion for use with peridynamic states. *International Journal for Multiscale Computational Engineering* **9**, 675–687 (2011).

44. Foster, J. T., Silling, S. A. & Chen, W. W. Viscoplasticity using peridynamics. *International journal for numerical methods in engineering* **81**, 1242–1258 (2010).
45. GEIGER, G. CERAMICS COATINGS: ENHANCE MATERIAL PERFORMANCE. *American Ceramic Society Bulletin* **71**, 1470–1470 (1992).
46. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. *Neural message passing for quantum chemistry* in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), 1263–1272.
47. Gioia, G. & Ortiz, M. Delamination of compressed thin films. *Advances in applied mechanics* **33**, 119–192 (1997).
48. Gori, M., Monfardini, G. & Scarselli, F. *A new model for learning in graph domains* in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* **2** (2005), 729–734.
49. Gough, G., Elam, C., Tipper, G. & De Bruyne, N. The stabilisation of a thin sheet by a continuous supporting medium. *The Aeronautical Journal* **44**, 12–43 (1940).
50. Goyal, S., Srinivasan, K., Subbarayan, G. & Siegmund, T. On instability-induced debond initiation in thin film systems. *Engineering Fracture Mechanics* **77**, 1298–1313 (2010).
51. Grill, A. Diamond-like carbon: state of the art. *Diamond and related materials* **8**, 428–434 (1999).
52. Gullerud, A. S., Gao, X., Dodds, R. H. & Haj-Ali, R. Simulation of ductile crack growth using computational cells: numerical aspects. *Engineering Fracture Mechanics* **66**, 65–92 (2000).
53. Ha, Y. D. & Bobaru, F. Characteristics of dynamic brittle fracture captured with peridynamics. *Engineering Fracture Mechanics* **78**, 1156–1168 (2011).
54. Ha, Y. D. & Bobaru, F. Studies of dynamic crack propagation and crack branching with peridynamics. *International Journal of Fracture* **162**, 229–244 (2010).
55. Harrison, C., Stafford, C. M., Zhang, W. & Karim, A. Sinusoidal phase grating created by a tunably buckled surface. *Applied Physics Letters* **85**, 4016–4018 (2004).
56. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.
57. Heaton, J., Polson, N. & Witte, J. H. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* **33**, 3–12 (2017).
58. Hillerborg, A., Mod  r, M. & Petersson, P.-E. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and concrete research* **6**, 773–781 (1976).

59. Hu, W., Ha, Y. D. & Bobaru, F. Modeling dynamic fracture and damage in a fiber-reinforced composite lamina with peridynamics. *Journal for Multiscale Computational Engineering* **9**, 707–726 (2011).
60. Hu, W., Ha, Y. D. & Bobaru, F. Peridynamic model for dynamic fracture in uni-directional fiber-reinforced composites. *Computer Methods in Applied Mechanics and Engineering* **217–220**, 247–261 (2012).
61. Hu, W., Ha, Y. D., Bobaru, F. & Silling, S. A. The formulation and computation of the nonlocal J-integral in bond-based peridynamics. *International journal of fracture* **176**, 195–206 (2012).
62. Hu, Y., De Carvalho, N. & Madenci, E. Peridynamic modeling of delamination growth in composite laminates. *Composite Structures* **132**, 610–620 (2015).
63. Hutchinson, J. W. & Suo, Z. Mixed mode cracking in layered materials. *Advances in applied mechanics* **29**, 63–191 (1991).
64. Hutchinson, J. & Evans, A. Mechanics of materials: top-down approaches to fracture. *Acta materialia* **48**, 125–135 (2000).
65. Kamilaris, A. & Prenafeta-Boldú, F. X. Deep learning in agriculture: A survey. *Computers and electronics in agriculture* **147**, 70–90 (2018).
66. Khang, D.-Y., Jiang, H., Huang, Y. & Rogers, J. A. A stretchable form of single-crystal silicon for high-performance electronics on rubber substrates. *Science* **311**, 208–212 (2006).
67. Kilic, B., Agwai, A & Madenci, E. Peridynamic theory for progressive damage prediction in center-cracked composite laminates. *Composite Structures* **90**, 141–151 (2009).
68. Kilic, B & Madenci, E. An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics* **53**, 194–204 (2010).
69. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
70. Kipf, T., Fetaya, E., Wang, K.-C., Welling, M. & Zemel, R. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687* (2018).
71. Komvopoulos, K., Saka, N & Suh, N. P. The mechanism of friction in boundary lubrication. *Journal of Tribology* **107**, 452–462 (1985).
72. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems* (2012), 1097–1105.
73. Kunin, I. Elastic media with microstructure. I-One-dimensional models. *Springer Verlag Springer Series on Solid State Sciences* **26** (1982).

74. Kunin, I. Elastic media with microstructure. II-Three-dimensional models. *Springer Verlag Springer Series on Solid State Sciences* (1983).
75. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436–444 (2015).
76. Lee, J.-G. *et al.* Deep learning in medical imaging: general overview. *Korean journal of radiology* **18**, 570–584 (2017).
77. Lee, Y., Wetzels, E. & Wagner, N. J. "The ballistic impact characteristics of Kevlar® woven fabrics impregnated with a colloidal shear thickening fluid". *Journal of materials science* **38**, 2825–2833 ((2003)).
78. Lehoucq, R. B. & Silling, S. A. Force flux and the peridynamic stress tensor. *Journal of the Mechanics and Physics of Solids* **56**, 1566–1577 (2008).
79. Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).
80. Lifshitz, Y. Hydrogen-free amorphous carbon films: correlation between growth conditions and properties. *Diamond and Related Materials* **5**, 388–400 (1996).
81. Lin, A. & Meyers, M. A. "Growth and structure in abalone shell". *Materials Science and Engineering: A* **390**, 27–41 ((2005)).
82. Lipton, R. Dynamic brittle fracture as a small horizon limit of peridynamics. *Journal of Elasticity* **117**, 21–50 (2014).
83. Liu, W. & Hong, J.-W. Discretized peridynamics for brittle and ductile solids. *International Journal for Numerical Methods in Engineering* **89**, 1028–1046 (2012).
84. Lu, W & Komvopoulos, K. Nanotribological and nanomechanical properties of ultra-thin amorphous carbon films synthesized by radio frequency sputtering. *Journal of Tribology* **123**, 641–650 (2001).
85. Luan, B. & Robbins, M. O. The breakdown of continuum models for mechanical contacts. *Nature* **435**, 929–932 (2005).
86. Lubineau, G., Azdoud, Y., Han, F., Rey, C. & Askari, A. A morphing strategy to couple non-local to local continuum mechanics. *Journal of the Mechanics and Physics of Solids* **60**, 1088–1102 (2012).
87. Macek, R. W. & Silling, S. A. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design* **43**, 1169–1178 (2007).
88. Madenci, E. & Oterkus, E. *Peridynamic Theory and Its Applications* (Springer, 2014).
89. Madenci, E. & Oterkus, S. Ordinary state-based peridynamics for plastic deformation according to von Mises yield criteria with isotropic hardening. *Journal of the Mechanics and Physics of Solids* **86**, 192–219 (2016).
90. Marshall, D. B. & Evans, A. G. Measurement of adherence of residually stressed thin films by indentation. I. Mechanics of interface delamination. *Journal of Applied Physics* **56**, 2632–2638 (1984).

91. *Material Match* <https://matmatch.com>. Accessed: 2020-03-10. 2020.
92. Medvedovski, E. Ballistic performance of armour ceramics: Influence of design and structure. Part 1. *Ceramics International* **36**, 2103–2115 (2010).
93. Mei, H. Fracture and delamination of elastic thin films on compliant substrates: modeling and simulations (2011).
94. Mei, H., Landis, C. M. & Huang, R. Concomitant wrinkling and buckle-delamination of elastic thin films on compliant substrates. *Mechanics of Materials* **43**, 627–642 (2011).
95. Mitchell, A. R. & Griffiths, D. F. *The finite difference method in partial differential equations* (John Wiley, 1980).
96. Mitchell, J. A. A non-local, ordinary-state-based viscoelasticity model for peridynamics. *Sandia National Lab Report* **8064**, 1–28 (2011).
97. Mitchell, J. A. A nonlocal, ordinary, state-based plasticity model for peridynamics. *SAND report* **3166** (2011).
98. Murphy, K. P. *Machine learning: a probabilistic perspective* (MIT press, 2012).
99. National standard-0101.04. "Ballistic Resistance of Personal Body Armor" (2001).
100. Oishi, A. & Yagawa, G. Computational mechanics enhanced by deep learning. *Computer Methods in Applied Mechanics and Engineering* **327**, 327–351 (2017).
101. Oterkus, E. & Madenci, E. *Ordinary state-based peridynamic material constants in 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA* (2012), 1946.
102. Oudet, E. Approximation of Partitions of Least Perimeter by  $\Gamma$ -Convergence: Around Kelvin's Conjecture. *Experimental Mathematics* **20**, 260–270. eprint: <http://dx.doi.org/10.1080/10586458.2011.565233>. <http://dx.doi.org/10.1080/10586458.2011.565233> (2011).
103. Parks, M. L., Lehoucq, R. B., Plimpton, S. J. & Silling, S. A. Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications* **179**, 777–783 (2008).
104. Parks, M. L., Littlewood, D. J., Mitchell, J. A. & Silling, S. A. Peridigm Users' Guide v1. 0.0. *SAND Report* **7800** (2012).
105. Parks, M. L., Seleson, P., Plimpton, S. J., Silling, S. A. & Lehoucq, R. B. *Peridynamics with LAMMPS: A user guide v0.3 beta* Sandia Report 2011-8523, 2010. <http://prod.sandia.gov/techlib/access-control.cgi/2011/118523.pdf>.
106. Pian, T. H. Mechanical sublayer model for elastic-plastic analyses. *Computational mechanics* **2**, 26–30 (1987).

107. Raissi, M., Yazdani, A. & Karniadakis, G. E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030 (2020).
108. Reif, J. H. & Tate, S. R. *The Complexity of N-Body Simulation* in *20TH ANNUAL COLLOQUIUM ON AUTOMATA, LANGUAGES AND PROGRAMMING (ICALP'93)* (1993), 162–176.
109. Rice, J. R. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of applied mechanics* **35**, 379–386 (1968).
110. Rice, J. Elastic fracture mechanics concepts for interfacial cracks. *Journal of applied mechanics* **55**, 98–103 (1988).
111. Rockey, K., Evans, H., Griffiths, D. & Nethercot, D. *The finite element method* (Crosby Lockwood Staples London, 1975).
112. Rogula, D. in *Nonlocal theory of material media* 123–222 (Springer, 1982).
113. Sadanandan, S. & Hetherington, J. "Characterisation of ceramic/steel and ceramic/aluminium armours subjected to oblique impact". *International journal of impact engineering* **19**, 811–819 ((1997)).
114. Samal, M., Seidenfuss, M & Roos, E. A new mesh-independent Rousselier's damage model: Finite element implementation and experimental verification. *International Journal of Mechanical Sciences* **51**, 619–630 (2009).
115. Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development* **3**, 210–229 (1959).
116. Sanchez-Gonzalez, A. *et al.* Graph networks as learnable physics engines for inference and control. *arXiv preprint arXiv:1806.01242* (2018).
117. Sarikaya, M., Gunnison, K., Yasrebi, M. & Aksay, I. "Mechanical property-microstructural relationships in abalone shell" in *MRS Proceedings* **174** ((1989)), 109.
118. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks* **20**, 81–102 (2008).
119. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks* **20**, 81–102 (2008).
120. Scarselli, F. *et al.* Graph neural networks for ranking web pages in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)* (2005), 666–672.
121. Schwarzer, M. *et al.* Learning to fail: Predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks. *Computational Materials Science* **162**, 322–332 (2019).
122. Seleson, P., Beneddine, S. & Prudhomme, S. A force-based coupling scheme for peridynamics and classical elasticity. *Computational Materials Science* **66**, 34–49 (2013).



123. Seleson, P., Parks, M. L., Gunzburger, M. & Lehoucq, R. B. Peridynamics as an upscaling of molecular dynamics. *Multiscale Modeling & Simulation* **8**, 204–227 (2009).
124. Shield, T., Kim, K. & Shield, R. The buckling of an elastic layer bonded to an elastic substrate in plane strain. *Journal of Applied Mechanics* **61**, 231–235 (1994).
125. Silling, S. A. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* **48**, 175–209 (2000).
126. Silling, S. & Lehoucq, R. Peridynamic theory of solid mechanics. *Advances in Applied Mechanics* **44**, 73–168 (2010).
127. Silling, S. A. & Askari, E. A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures* **83**, 1526–1535 (2005).
128. Silling, S. A., Epton, M., Weckner, O., Xu, J & Askari, E. Peridynamic states and constitutive modeling. *Journal of Elasticity* **88**, 151–184 (2007).
129. Silling, S. A. & Lehoucq, R. B. Convergence of peridynamics to classical elasticity theory. *Journal of Elasticity* **93**, 13–37 (2008).
130. Silling, S. A. & Bobaru, F. Peridynamic modeling of membranes and fibers. *International Journal of Non-Linear Mechanics* **40**, 395–409 (2005).
131. Sinha, S., Ebrahimi, S. & Darrell, T. *Variational adversarial active learning* in *Proceedings of the IEEE International Conference on Computer Vision* (2019), 5972–5981.
132. Sternberg, J. “Material properties determining the resistance of ceramics to high velocity penetration”. *Journal of applied physics* **65**, 3417–3424 (1989).
133. Sukhbaatar, S., Fergus, R., *et al.* *Learning multiagent communication with backpropagation* in *Advances in neural information processing systems* (2016), 2244–2252.
134. TACC. *Texas Advanced Computing Center (TACC), The University of Texas at Austin* 2015. <https://portal.xsede.org/tacc-stampede>.
135. Taylor, M. J. *Numerical simulation of thermo-elasticity, inelasticity and rupture in membrane theory* PhD thesis (University of California, Berkeley, 2008).
136. Van Steenkiste, S., Chang, M., Greff, K. & Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353* (2018).
137. Volinsky, A. A., Moody, N. R. & Gerberich, W. W. Interfacial toughness measurements for thin films on substrates. *Acta Materialia* **50**, 441–466 (2002).
138. Warren, T. L. *et al.* A non-ordinary state-based peridynamic method to model solid material deformation and fracture. *International Journal of Solids and Structures* **46**, 1186–1195 (2009).
139. Wilkins, M., Honodel, C. & Sawle, D. *"APPROACH TO THE STUDY OF LIGHT ARMOR"* tech. rep. (California Univ., Livermore. Lawrence Radiation Lab., (1967)).

- 140. Xia, L. & Shih, C. F. Ductile crack growth-I. A numerical study using computational cells with microstructurally-based length scales. *Journal of the Mechanics and Physics of Solids* **43**, 233–259 (1995).
- 141. Xia, L. & Shih, C. F. Ductile crack growth—II. Void nucleation and geometry effects on macroscopic fracture behavior. *Journal of the Mechanics and Physics of Solids* **43**, 1953–1981 (1995).
- 142. Xie, T. & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters* **120**, 145301 (2018).
- 143. Xu, J., Askari, A., Weckner, O. & Silling, S. Peridynamic analysis of impact damage in composite laminates. *Journal of Aerospace Engineering* **21**, 187–194 (2008).
- 144. Zienkiewicz, O., Owen, D. & Nayak, G. *Composite and 'overlay' models in numerical analysis of elasto-plastic continua* in *International Symposium on Foundations of Plasticity, Warsaw, Poland* (1973), 107–123.
- 145. Zienkiewicz, O. C., Taylor, R. L., Zienkiewicz, O. C. & Taylor, R. L. *The finite element method* (McGraw-hill London, 1977).