

Multi-scale Modelling of Ice-Structure Interactions

Ph.D. Thesis

Bozo Vazic

Department of Naval Architecture, Ocean, and Marine Engineering
University of Strathclyde, Glasgow
United Kingdom

July 15, 2020

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: Bozo Vazic

Date: July 15, 2020

Abstract

The present investigation has been devoted to the modeling of the ice-structure interactions by using the state-of-the-art technique, peridynamics. Peridynamics is a new continuum mechanics formulation originally developed at Sandia National Laboratories, USA and very suitable for failure analysis of structures due to its mathematical structure. Structures can vary from thin-walled structures such as ship hulls or airplane fuselage to bridges and wind-turbines. Furthermore by using peridynamics structural failure can be observed as compressive, tensile, bending or buckling failure and materials can be classified as elastic, viscoelastic or plastic. Peridynamic equation is in integro-differential equation form rather than a partial differential equation as in the classical continuum mechanics which allows the continuous usage of these equations without specially treating the discontinuities. Although relatively new, it is successfully verified and utilized for modelling both metallic and composite structures. Hence, it is an excellent candidate to investigate complex problems such as the ice-structure interaction modelling. Furthermore, in a general sense, it may bring a new dimension to the analysis of marine structures especially in the area of arctic engineering. Furthermore, peridynamic solver was developed including 2D and 3D geometry definitions together with peridynamic mesh. Several different solvers were implemented, such as explicit solver, adaptive dynamic relaxation and direct solver. In order to reduce the computational time, several family search algorithms (such as brute-force search, region partitioning algorithm, K-d tree and R-tree algorithms) were tested and implemented together with parallelization of most time consuming parts of code. Finally, several numerical studies were considered in order to demonstrate ice-structure interaction via peridynamic analysis. Where those numerical studies range from 2D and 3D Bond

Based peridynamic models used for analysis of ice splitting loads for in-plane failure and impact analysis between cylindrical (offshore structures) and conical (ship's bow) rigid bodies and ice sheet. Furthermore advanced peridynamic model for Mindlin plate resting on Winkler foundation was developed in order to test for out-of-plane failure of an ice sheet.

Preface/Acknowledgements

First of all, I would like to thank Prof. Erkan Oterkus for all the help and guidance during my PhD.

I'm also very grateful to my wife who was beside me every step of the way.

Next I would like to thank my parents and my brother for all the support.

Lastly, I would like to express my gratitude to University of Strathclyde and Lloyd's Register for their sponsorship.

Table of Contents

List of Figures	i
List of Tables	vii
Nomenclature	ix
1 Introduction	1
2 Ice Material Basics	6
2.1 Introduction	6
2.2 Crystallographic Structure of Hexagonal Ih Ice	7
2.3 Land Based Glacial Ice	8
2.4 Sea Ice	10
2.4.1 First-Year Ice	11
2.4.2 Multi-Year Ice	13
2.5 Mechanical Properties of Ice	14
2.5.1 Introduction	14
2.5.2 Elastic Properties	15
2.5.3 Uniaxial Strength	19
2.5.4 Shear strength	28
2.5.5 Fracture Toughness	31
3 Peridynamic Theory	35
3.1 Introduction	35
3.2 Basics of Peridynamic Theory	38

3.2.1	Ordinary State Based Peridynamics	44
3.2.2	Bond Based Peridynamics	48
3.2.3	Introducing damage into Peridynamics	51
4	Peridynamic Solver - VOPDSolver	56
4.1	Introduction	56
4.2	Geometry and Mesh	57
4.2.1	Geometry Definition	58
4.2.2	Mesh Definition	60
4.2.3	Voronoi Tesselation for Peridynamic Mesh	66
4.3	Family Search	68
4.3.1	Brute-force search	70
4.3.2	Region Partitioning	71
4.3.3	Tree data structures	76
4.4	Solver methods	91
4.4.1	Dynamic Solver	92
4.4.2	Static and Quasi-static Solver	93
4.4.3	Parallel Execution	98
4.5	Impact Definition	99
5	Examples	105
5.1	Introduction	105
5.2	Dynamic propagation of a macrocrack interacting with parallel small cracks	109
5.2.1	Introduction	109
5.2.2	Problem Definition	110
5.2.3	Numerical Results	110
5.2.4	Conclusion	126
5.3	Peridynamic Model for a Mindlin Plate Resting on A Winkler Elastic Foundation	126
5.3.1	Indroduction	126

5.3.2	Peridynamic Mindlin Plate Formulation	128
5.3.3	Direct Solution of the Peridynamic Mindlin Plate Formulation . .	130
5.3.4	Peridynamic Mindlin Plate Resting on An Elastic Foundation . .	133
5.3.5	Numerical Results	135
5.3.6	Conclusion	142
5.4	In-Plane and Out-Of Plane Failure of an Ice Sheet Using Peridynamics .	142
5.4.1	Introduction	142
5.4.2	2D Bond Based Peridynamic Model for In-Plane Failure of an Ice Sheet	145
5.4.3	Peridynamic Mindlin Plate on Winkler Foundation Model for Out-Of-Plane Failure of an Ice Sheet	148
5.4.4	Conclusion	151
5.5	Peridynamic Approach for Modelling Ice-Structure Interactions	152
5.5.1	Introduction	152
5.5.2	2D Bond Based Peridynamic Mindlin Plate on Winkler Foundation	153
5.5.3	3D Bond Based Peridynamic Model with Rigid Body Impact Load	155
5.5.4	Conclusion	159
5.6	Ice-structure Interaction Applied to the Lighthouse Example	160
5.6.1	Introduction	160
5.6.2	Problem Description	161
5.6.3	2D Bond Based Peridynamic Model of the Ice Sheet and the Lighthouse	163
5.6.4	Conclusion	165
5.7	Family Member Search Algorithms for Peridynamic Analysis	167
5.7.1	Introduction	167
5.7.2	Comparative Performance of Search Algorithms	167
5.7.3	Conclusion	170
6	Conclusion	172
6.0.1	Future Work	173

List of Figures

2.1	Left: Crystal structure of hexagonal crystalline Ih ice - red spheres represent oxygen atoms and black bars represent hydrogen atoms. Right: A schematic image of hexagonal ice defining main crystal faces and axis.	8
2.2	The Margerie Glacier at Glacier Bay National Park in Alaska	9
2.3	Density-pressure relationships at Byrd and Mizuho Stations in Antarctica. The pressure is the load due to the overburden snow, Maeno and Ebinuma (1983)	10
2.4	Pancake ice in the Barents Sea at the onset of winter. The size of the largest pancakes shown in this figure is roughly one meter, Notz (2005) .	12
2.5	Schematic representation of a cross-section of first-year sea ice, Sand (2008)	13
2.6	Forming of multi-year ice, Sand (2008)	14
2.7	Standard setup for uniaxial compression test on sea ice, Timco and Weeks (2010)	19
2.8	Saline-ice specimen with end caps, Cole et al. (1985)	20
2.9	Tensile strength as a function of porosity for horizontally loaded first year sea ice, Timco and Weeks (2010)	24
2.10	Transition between ductile and brittle failure of ice under compressive load, Sand (2008)	26

2.11	Beginning stages in the development of wing cracks under compressive loading. (a) Initial low stress develops tensile zones around the tips of the inclined cracks. (b) By increasing stress wing cracks initiate at the inclined crack tips. This is due to frictional sliding across inclined crack. (c) At even bigger stress, the wing cracks propagate and separate their faces due to further sliding, Schulson and Duval (2009)	27
2.12	Asymmetric four-point loading apparatus together with shear force and bending moment diagrams, Frederking and Timco (1986)	29
2.13	Three modes of crack propagation	32
2.14	Left image: Four-point-bend configuration and Right image: Three-point-bend configuration	32
2.15	Plane-strain fracture toughness vs. temperature, Schulson and Duval (2009)	33
3.1	The material point \mathbf{x} interacts with other material points inside its horizon H_x	37
3.2	The material point \mathbf{x} interacts with other material points inside its horizon H_x	39
3.3	Nonordinary State Based Peridynamics	44
3.4	Ordinary State Based Peridynamics	45
3.5	Bond Based Peridynamics	48
3.6	Bond force as a function of bond stretch	51
3.7	Interaction between points which cross the fracture surface	53
3.8	Integration domain of the micropotentials crossing a fracture surface . .	54
4.1	Rough peridynamic mesh of a ship hull cross section; a) 2D and b) 3D .	58
4.2	2D peridynamic mesh generation	59
4.3	VOPDGeometry 2D class diagram	59
4.4	VOPDGeometry 3D class diagram	60
4.5	VOPDMesh class diagram	62
4.6	Basic idea behind Ray Casting Algorithm	63

4.7	Polyhedron	64
4.8	Determining if the PD point is inside or outside the polyhedron	64
4.9	Inner triangle vs polyhedron face	66
4.10	2D Voronoi tessellation	67
4.11	3D Peridynamic mesh with Voronoi tessellation	68
4.12	Brute-force search algorithm for the material points, $\mathbf{x}_{(i)}$	71
4.13	Material points inside the domain of interest with square grids	72
4.14	Partitioned rectangular problem domain	73
4.15	The neighboring regions for the material point 109	75
4.16	General concept of a tree structure	77
4.17	Two types of binary tree; a) full- and b) complete-tree structures	78
4.18	Balanced versus unbalanced binary tree; a) balanced- and b) unbalanced	78
4.19	Binary Search Tree (BST) with left and right subtrees	79
4.20	First two levels of R-tree	81
4.21	First two levels of K-d tree	82
4.22	Intersecting MBRs, where peridynamic family member points are only in MBR A	83
4.23	MBRs data and their inner nodes	84
4.24	MBRs data and their inner nodes	85
4.25	2-d tree	86
4.26	Data slicing for 3-d tree	88
4.27	A K-d tree built from (x, y, z) tuples	89
4.28	Solver selection flowchart	92
4.29	Using Eigen C++ library to solve linear system of equations	98
4.30	Example of for loop parallelization with openMP	99
4.31	Rigid impactor and deformable body interaction (Madenci and Oterkus (2014))	100
4.32	Impact of two identical flexible bars - displacement predictions in the x-direction at the centers $(\pm 0.025, 0.0, 0.0)$ of the left and right bars as time progresses for peridynamics and FEA	102

4.33	A rigid disk impacting on a rectangular plate peridynamic model	104
5.1	Problem definition	111
5.2	Peridynamic model and its discretization	111
5.4	Collinear small crack in front of a macrocrack	112
5.3	Macrocrack propagation without considering small cracks, (a) 4×10^{-5} s, (b) 6.8×10^{-5} s	113
5.5	Macrocrack propagation for different values of a/b (a) $a/b = 0.2$, (b) $a/b = 0.4$, (c) $a/b = 0.6$, (d) $a/b = 0.8$ at 6.4×10^{-5} s	114
5.6	Two parallel small cracks	115
5.7	Comparison of a single macrocrack and macrocrack with two parallel small cracks (a) macrocrack without a small crack, (b) $h/l = 0.75$, $s/l = 2$, (c) $h/l = 1$, $s/l = 2$, (d) $h/l = 1.25$, $s/l = 2$ at 6×10^{-5} s	116
5.8	Crack propagation for $h/l = 1.25$ (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s	117
5.9	Set of horizontal small cracks	118
5.10	Crack propagation of horizontal small cracks (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s	119
5.11	Set of vertical small cracks	120
5.12	Crack propagation of vertical small cracks (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s	121
5.13	Set of vertical small cracks	122
5.14	(a) Initial configuration and (b) crack propagation in a set with thirty-two small cracks	123
5.15	Set of eighty small cracks	124
5.16	(a) Initial configuration and (b) crack propagation in a set with eighty small cracks	125
5.17	Initial and deformed configuration of a Mindlin plate (Diyaroglu et al. (2015))	128
5.18	Mindlin plate on a Winkler foundation	134

5.19	Peridynamic discretization of Mindlin plate subjected to the transverse loading (shaded area)	136
5.20	FEA results for displacement w	136
5.21	FEA results for and rotations (a) ϕ_x and (b) ϕ_y	137
5.22	Peridynamic Mindlin plate results for displacement w	138
5.23	Peridynamic Mindlin plate results for rotations (b) ϕ_x and (c) ϕ_y	139
5.24	(a) Pre-cracked Mindlin plate under pure bending condition, (b) Peridynamic discretization of pre-cracked Mindlin plate resting on an elastic Winkler foundation	141
5.25	Crack propagation for PMMA pre-cracked plate resting on a Winkler foundation	142
5.26	Contact and load description	145
5.27	Illustration of the model utilized to study in-plane failure of ice plate	146
5.28	Splitting load comparison between PD results and experiments	147
5.29	Model outline for out-of-plane failure of ice sheet	149
5.30	Comparison between Kerr's relation and PD results for different size semi-infinite plates	151
5.31	Illustration of the model utilised to study ice floe failure	154
5.32	Ice failure for $n = 4,5,6,7$, and 9	156
5.33	Plate rotation for $L = 1 \times l$	157
5.34	3D Model	157
5.35	Ice fracture behaviour	157
5.36	Ice fracture at time step of 30000	158
5.37	Ice fracture at time step of 36000	159
5.38	Ice fracture at time step of 40000	159
5.39	Norströmsgrund lighthouse (Bjerkas et al. (2009))	161
5.40	Lighthouse Norströmsgrund full-scale measurement set-up (Bjerkas et al. (2009), Bjerkås (2006))	162
5.41	Parametrization of the lighthouse model	163

5.42	Lighthouse Norströmsgrund full-scale measurement set-up (Bjerkas et al. (2009), Bjerkås (2006))	165
5.43	Simulated horizontal global ice force on the lighthouse	166
5.44	Simulated horizontal global ice force on the lighthouse compared to measured loads (Bjerkas et al. (2009))	166
5.45	Time comparison for different family search algorithms	168
5.46	Time comparison for different horizon sizes $H = n\Delta$ for 1000000 points	169
5.47	Time comparison for building the tree structure between Balanced K-d tree and Boost R-tree with packing algorithm	171

List of Tables

2.1	Polycrystalline Ih ice (Isotropic) average values at -16°C , Gammon et al. (1983)	17
2.2	Calculated Young's and shear modulus of homogeneous orthotropic sheets of ice at -16°C , Schulson and Duval (2009)	17
2.3	Calculated Poisson's ratio of homogeneous orthotropic sheets of ice at -16°C , Schulson and Duval (2009)	18
2.4	The tensile strength of columnar-grained first-year sea ice loaded uniaxially across the columns	22
2.5	Shear strength of columnar-grained sea ice. Grain orientation is measured relative to shear plane Frederking and Timco (1986).	30
4.1	Example structures of trees created by different algorithms and their operations times*	90
5.1	Macrocrack propagation speed at 4×10^{-5} s	112
5.2	Macrocrack propagation speed at 4×10^{-5} s	115
5.3	Overall crack propagation length of horizontal small cracks at 4×10^{-5} s	118
5.4	Crack propagation speed of vertical small cracks at 4×10^{-5} s	120
5.5	Test model dimensions and peridynamic inputs	146
5.6	Problem setup parameters	158
5.7	Event from the morning of March 30 th 2003 (Bjerkas et al. (2009))	163
5.8	Material properties for ice	164
5.9	Family search time comparison	168

5.10 Times necessary for building the tree structure	170
--	-----

Nomenclature

List of Symbols

$\ddot{\mathbf{U}}$	Acceleration matrix
$\ddot{\mathbf{u}}$	Acceleration vector
Δt	Time step size
Δx	Distance between material points
δ	Radius of a domain of influence
$\Delta \xi$	Spacing between two consecutive material points
$\dot{\mathbf{u}}$	Velocity vector
η	Relative displacement vector of two material points
\hat{b}	Body load arising from the shear deformation
\hat{f}	Pairwise force function related with the shear deformation
κ	Bulk Modulus
κ_c	Critical curvature
ξ	Relative position vector of two material points in the reference configuration
\mathbf{B}	Body load matrix

b	Body load vector
C	Artificial damping matrix
D	Fictitious density matrix
E	Young's modulus
F	Total force acting on a rigid body
f	Pairwise force function
K	Stiffness matrix
M	Artificial mass matrix
P	Vector of internal forces
t'	Force density vector of material point \mathbf{x}'
t	Force density vector of material point \mathbf{x}
U	Displacement matrix
u	Displacement vector
x'	Position vector of the member material point
x	Position vector of the main material point
y'	Deformed position vector of the member material point
y	Deformed position vector of the main material point
μ	Shear Modulus
ν	Poissons ratio
ω	Influence function
ϕ	Shear rotation of a material point

ϕ_c	Critical shear angle
ρ	Density
θ	Dilatation
φ	Orientation of a bond with respect to x axis
\tilde{b}	Body load arising from the bending deformation
\tilde{f}	Pairwise force function related with the bending deformation
ξ	Distance between the material points
A	Cross-sectional area
c	Bond constant
D	Flexural rigidity
F_n	Contact force
G	Energy release rate
G_0	Critical energy release rate
G_I	Mode I energy release rate
G_{Ic}	Mode I critical energy release rate
G_{IIc}	Mode II critical energy release rate
G_{IIIc}	Mode III critical energy release rate
G_{III}	Mode III energy release rate
G_{II}	Mode II energy release rate
h	Thickness of the structure
H_x	The domain of influence

j	Member material point
K	Stress intensity factor
k	Main material point
K_c	Critical stress intensity factor
K_{Ic}	Mode I critical stress intensity factor
K_{IIc}	Mode II critical stress intensity factor
K_{IIIc}	Mode III critical stress intensity factor
L	Lagrangian
s	Stretch
s_0	Critical stretch
$sFac$	Safety factor
T	Kinetic energy
t	Time
U	Potential energy
V	Volume of a material point
W	Strain energy density function
w	Micropotential of a bond

List of Abbreviations

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

ADR	Adaptive Dynamic Relaxation
CCM	Classical Continuum Mechanics
CMOD	Crack Mouth Opening Displacement
CPU	Central Processing Unit
CTOA	Crack Tip Opening Angle
CTOD	Crack Tip Opening Displacement
CZE	Cohesive Zone Elements
DR	Dynamic Relaxation
EOM	Equation of Motion
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
FSI	Fluid Structure Interaction
GPU	Graphics Processing Unit
HPC	High Performance Computing
LEFM	Linear Elastic Fracture Mechanics
MIMD	Multiple Instruction Multiple Data
MISD	Multiple Instruction Single Data
PD	Peridynamic
PINN	Physics Informed Neural Network
PMMA	Polymethyl methacrylate

SIMD	Single Instruction Multiple Data
XFEM	Extended Finite Element Method

Chapter 1

Introduction

Due to predicted oil and gas reserves Arctic is considered to become as important as Middle East in the near future. Around 30% of the world's undiscovered gas and 13% of the world's undiscovered oil are expected to be stored in the North Arctic Circle, Bird et al. (2008). Furthermore, as the ice-covered region size is decreasing as a result of the climate change, sailing routes from Western Europe via Arctic to East Asia can be a reality and reduce the distance of traditional routes by one-third. If this happens, the Arctic can be one of the world's busiest sailing areas.

Despite of its advantages, utilization of the Arctic region for sailing brings new challenges due to its harsh environment. Therefore, ship structures must be designed to withstand ice loads in case of a collision between a ship and ice takes place. Such incidents can cause significant damage on the structure which can yield flooding and sinking of the ship. After the Titanic disaster, collisions between ships and medium or big size icebergs have been rare due to high-tech radar and satellite systems. However, it is still difficult to detect small size icebergs and collisions with such small size icebergs can still cause severe damage. From mid 1800 till late 2000 there have been around 23 ships that sunk as a result of collision with an iceberg, with the last one being MS Explorer in 2007. On the other hand there have been numerous collisions with smaller icebergs that usually result in less damage, Hill (2009).

Although experimental studies can give invaluable information about ship-ice interactions, full scale tests are very costly to perform. Therefore, computer simulations can

be a good alternative. Ice-structure interaction modelling is a very challenging process. First of all, ice material response depends on many different factors including applied-stress, strain-rate, temperature, grain-size, salinity, porosity and confining pressure. Furthermore, macro-scale modeling may not be sufficient to capture the full physical behaviour because the micro-scale effects may have a significant effect on macroscopic material behaviour. Hence, it is necessary to utilize a multi-scale methodology.

In order to capture the macro-scale behaviour of ice, well-known Finite Element Method (FEM) has been used in various previous studies. Within FEM framework, various techniques can be used to model crack propagation such as cohesive zone models (CZM) and extended finite element method (XFEM). However, a universally accepted CZM failure model is not currently available and the crack propagation may have mesh dependency. Although, the mesh dependency problem can be overcome by XFEM, enrichment process may lead to an algebraic system with billions of unknowns which is difficult to solve numerically. This stems from the primary problem that is related to the need of embedding an initial crack a priori into the FE mesh to trigger crack propagation. In order to insert cracks at appropriate locations raises the need for an expert user. This can be solved by determining critical region(s) which are more likely to fail on which than XFEM nodal enrichment needs to be applied. This means that without an expert user, enriching nodes of the entire domain of a FE mesh becomes a standard practice, which in turn results in more run-time and reduced computational efficiency, Elruby et al. (2018). Furthermore, FEM is based on classical continuum mechanics which does not have a length scale parameter and is incapable of capturing phenomenon at the micro-scale. Hence, other techniques should be utilized at the micro-scale and linked to FEM simulation. However, it is not straightforward to obtain a smooth transition between different approaches at different scales.

By taking into account all these challenging issues, a state-of-the-art technique, peridynamics will be utilized for ice-structure interaction modelling. Peridynamics is a non-local continuum mechanics formulation which is very suitable for failure analysis of materials due its mathematical structure. Cracks can occur naturally in the formulation and there is no need to impose an external crack growth law, such as McClintock

equation where an upper limit of crack growth is based on the cyclic crack tip opening displacement or Paris-Erdogan Equation which models the relationship between the velocity of crack propagation and an abstract quantity called the range of stress intensity, which describes the magnitude of the stress at the crack tip. Furthermore, due to its non-local character, it can capture the phenomenon at multiple scales. Since same type of equations will be used at different scales, a smooth transition is expected. This means that standard peridynamic multiscale method for analysing defects has the following procedure where each level is a separate model with its own horizon. Each level interacts only with levels immediately above and below it. Smallest level/horizon is defined at the crack tip and other levels occupy bigger and bigger region with larger horizons. Equation of motion is applied only within each level. Higher levels provide boundary conditions on lower levels. Lower levels provide coarsened material properties (including damage) to higher levels. This means that in principle, a large number of levels can be used, all coupled in the same way.

The main aim of this proposed PhD study is to bring a new dimension contrary to the existing analytical and numerical modelling tools used for ice-structure interaction modelling. Specific objectives of the proposed study and the work program can be listed as:

- Development of a basic peridynamic code suitable for the problem of interest
- Perform a validation study against benchmark problems
- Implement contact analysis capability within the peridynamic code to represent ice-structure interaction.
- Perform a validation study for the contact analysis
- Compare the results against available data found in the literature including experimental, analytical and numerical studies
- Investigate ways to reduce the computational time by using shared memory multiprocessing

Chapter 2 starts with short overview of crystallographic structure of hexagonal Ih ice which is also called ordinary ice as it is basis for all ice on earth, followed by brief discussion on formation of land based glacial ice and sea ice. Furthermore, formation of first-year and multi-year ice is also discussed. This is followed by description of fundamental mechanical properties and behaviour of both columnar and granular ice. Subjects discussed are the elastic properties of ice, uniaxial strength, shear strength and fracture toughness.

Chapter 3 contains a short review of peridynamic theory. In this review strain energy formulation is explained which leads to development of peridynamic equation of motion. This is followed by detailed explanation of two basic peridynamic formulations - bond based and ordinary state based. At the end damage is introduced into the peridynamic framework.

Chapter 4 gives an in depth review of VOPDSolver that was developed as a part of this PhD study. VOPDSolver is an object oriented peridynamic solver that has ability to define complex geometrical models, apply different loading conditions, perform spatial searches and conduct impact analysis. First geometry and mesh definitions are presented. In geometry section overview of classes for 2D and 3D objects are presented. Mesh Definition section includes explanation of algorithms necessary for mesh definition, such as ray casting method for 2D and point to plane distance algorithm for 3D mesh. Furthermore, detailed description of family search algorithms is addressed, such as brute-force search, region partitioning algorithms and variety of tree data structures. Different solver methods are explained - dynamic solver and static/quasi-static solver, together with their parallel execution using openMP. At the end impact algorithms are introduced.

Chapter 5 contains examples that show the utilization of VOPDSolver in order to study ice-structure interactions. First study is by Vazic et al. (2017) where an in-depth analysis of interaction between macrocrack and parallel microcracks is presented. Second study is on Mindlin plate resting on Winkler foundation (Vazic et al. (2019c)) which shows applicability of Winkler foundation when modeling fluid-structure interactions applied to a peridynamic formulation of a Mindlin plate. Next study is on

in-plane and out-of-plane failure of an ice floe, where it is shown that peridynamics can be used to obtain good agreement with experiments for in-plane failure splitting load and prove Kerr's relation (Kerr (1976)) for out-of-plane failure of semi-infinite plate. Furthermore, study on ice floe fracture patterns for 2D and 3D models was done (Vazic et al. (2019b)). Next a study on interaction between a lighthouse and an ice floe was conducted where it was shown that peridynamics can give a good prediction of ice loads on offshore structures. Finally, a study on family member search algorithms (Vazic et al. (2019a)) was done in order to find most suitable algorithm which will have the biggest influence on decreasing computational time needed for family search.

Chapter 2

Ice Material Basics

2.1 Introduction

Term that describes parts of Earth's surface where water is frozen is called cryosphere. Ice has an important role in the global climate, especially regarding to water cycle. Ice can be found in the form of ice sheet, ice caps, ice fields or glaciers on land, ice sheets on lakes and as sea ice in oceans. There are 18 known solid crystalline forms of ice at ordinary pressures. The stable phase is called ice I, with two closely related low-pressure variants: hexagonal ice Ih and cubic ice Ic. Ice Ih is termed ordinary ice and it's the basis for all natural snow and ice on Earth. Ih ice has six-fold symmetry reflected in the shape of snow-flakes. On the other hand ice Ic is a metastable ice crystal that is made by depositing water vapor at ambient pressure with temperatures lower than 130°C. Throughout this chapter we are only going to deal with hexagonal Ih ice as it is the basis for naturally-occurring morphological forms - snow, firn (multi-year snow), freshwater ice and sea ice.

When dealing with marine environment, only two different ice types can be found, that is, sea ice and land based glacier ice in the form of icebergs (Sand (2008)). Sea ice is formed by cooling and freezing of sea water, whereas icebergs are fragments of land based glacial ice formed by large accumulation of snow which, through years of compression and freezing, turns into ice. In-depth discussion on growth, structure, and properties of sea ice can be found in Weeks and Ackley (1986) and Schulson and Duval

(2009). Thermal and mechanical history will determine micro-structure of ice that can be found in the marine environment. This will then have a strong influence on physical and mechanical properties of ice.

2.2 Crystallographic Structure of Hexagonal Ih Ice

One of the interesting characteristics of Ih ice is its density. Ordinary Ih ice is less dense than its melt. This means that ice floats on the surface of lakes and seas. If ice was more dense than its melt it would sink like as most solids do, this would then create a situation where natural water bodies would completely freeze up to their sea beds. This situation wouldn't be conducive to the development of marine life in either temperate or polar regions.

Molecular structure of ice is based on tetrahedral geometry of water molecule which leads to a highly ordered but loose structure, where each oxygen atom is surrounded by four hydrogen atoms. This tetrahedral coordination of the oxygen atoms will produce a crystal structure with hexagonal symmetry, a fact that will affect many of the large scale characteristics of ice (Weeks and Ackley (1986)). Because of a loose structure of ice, its density is less than in liquid state, where the ordered structure is partially broken down and the distance between water molecules is smaller on average. Hexagonal ice crystals form hexagonal plates and columns where the top and bottom faces are called basal planes see (Figure 2.1). The normal to the basal plane is referred to as the c-axis.

Looking at the hexagonal crystal structure of Ih ice one can see that the basic ice crystal growth form will be the hexagonal prism. Perfectly formed crystal of ice in this form would be difficult to deform as every possible bond in the crystal lattice have been made. In practice, however, the crystal lattice is formed with numerous faults, such as, point defect or line defects. Point defects are atomic-sized features that form within the ice lattice as:

Vacancies - empty molecular site

Interstitials - formed when a molecule becomes dislocated within the open space of the Ih crystal lattice

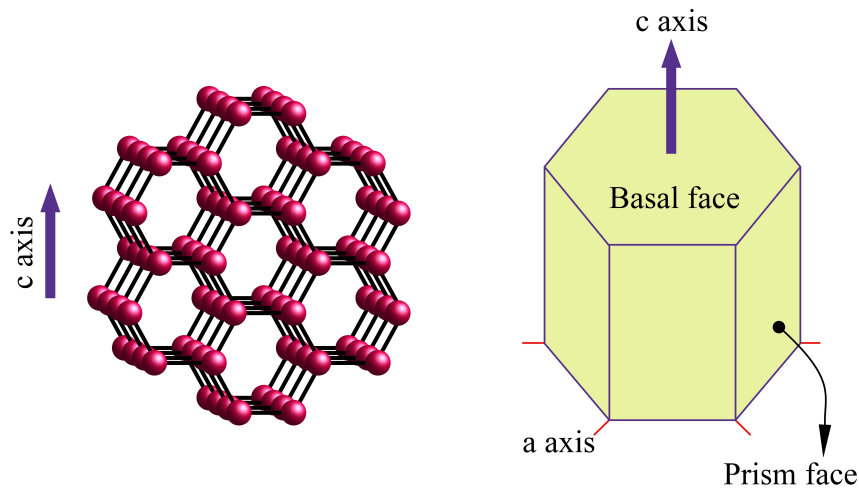


Figure 2.1: Left: Crystal structure of hexagonal crystalline Ih ice - red spheres represent oxygen atoms and black bars represent hydrogen atoms. Right: A schematic image of hexagonal ice defining main crystal faces and axis.

Solutes - impurities incorporated within the solid matrix, such as, brine pockets within sea ice

Ionic and Bjerrum Defects - violations of the Bernal-Fowler ice rules results in protonic defects

Line defects are dislocations within the crystal lattice which are created during the thermal-mechanical processes of ice crystal growth. These line defects are fundamental to plasticity and strength of ice. In addition to point and line defects, ice can have planar and volumetric defects. Planar defects relevant to fracture and creep are grain boundaries and free surfaces. Most common volumetric defects are pores, which are formed as the result of rejecting oxygen and nitrogen from water (Schulson and Duval (2009)).

2.3 Land Based Glacial Ice

Glacier is a persistent body of dense ice that is made up from compressed fallen snow. Glaciers form where the accumulation of snow and ice exceeds ablation (process



Figure 2.2: The Margerie Glacier at Glacier Bay National Park in Alaska

*(<https://www.nps.gov/glba/learn/kidsyouth/glaciers-of-glacier-bay-national-park.htm>)

opposite to accumulation - melting and sublimation) over many years, often centuries. What makes glaciers unique is their ability to flow due to stresses induced by their weight.

Main process behind densification of seasonal snow and its transformation into glaciers is sintering. Sintering by definition is a process of compacting and forming a solid mass of material either by heat or pressure without changing its physical phase. In case of glaciers sintering process is not only driven by excess of surface free energies in the system (heat) but also through external pressure. There are three distinct stages of densification of snow at polar glaciers. Those three stages are divided by two critical densities (Maeno and Ebinuma (1983)). At the beginning (first stage) snow will mainly densify through mechanical destruction and packing of snow crystals until the first critical density is reached, $\rho = 500\text{kg}/\text{m}^3$ (relative density $\rho_r = 0.60$). During second stage main mechanisms of densification are plastic deformation and

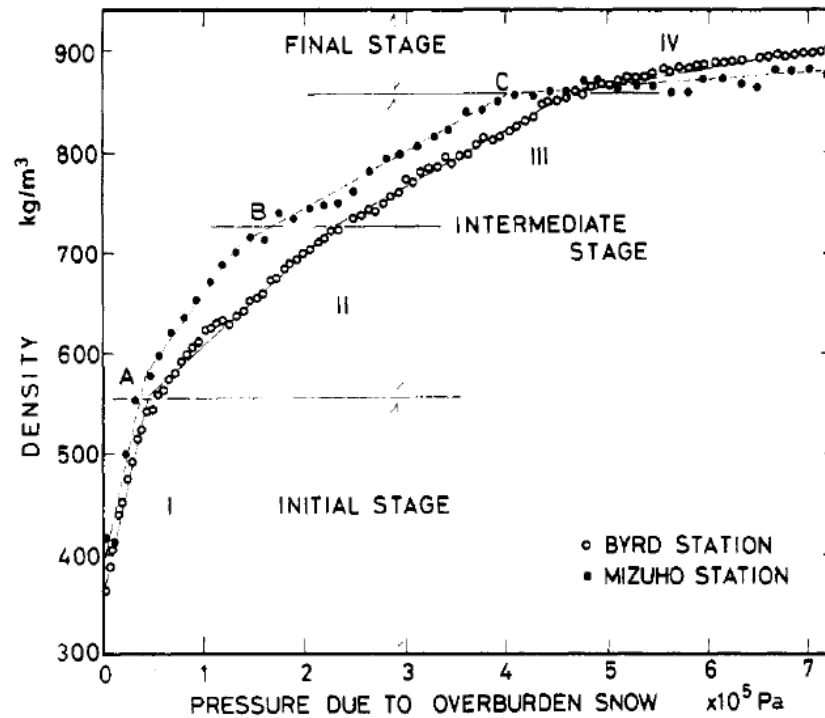


Figure 2.3: Density-pressure relationships at Byrd and Mizuho Stations in Antarctica. The pressure is the load due to the overburden snow, Maeno and Ebinuma (1983)

recrystallization until second critical density is reached, $\rho = 820 - 840 \text{ kg/m}^3$ ($\rho_r = 0.89 - 0.92$). Third and final stage is characterized by shrinkage of entrapped air bubbles until final/theoretical density of ice is reached, $\rho = 917 \text{ kg/m}^3$. These three stages can be shown on the pressure/density diagram measured in Byrd and Mizuho Stations in Antarctica, (see Figure 2.3)

2.4 Sea Ice

Sea ice is created when seawater freezes which is approximately at -1.9°C . As it was mentioned before ice floats because it is one of few substances where the solid phase is less dense than its liquid phase. About 7% of the Earth's surface is occupied by sea ice and about 12% of the world's oceans. Antarctic and Arctic polar ice packs encapsulate most of the world's sea ice. Surface of polar packs will undergo significant yearly cycling, which extensively affect Arctic ecology, including the ocean's ecosystems

(Wadhams (2003)).

Sea ice due to dynamic nature of marine environment (winds, currents and temperature fluctuations) can be found in variety of types and features. Furthermore sea ice can be classified based on age as:

New ice - recently frozen water

Nilas - sea ice crust up to 0.1 m

Young ice - transition phase between nilas and first-year ice, from 0.1 to 0.3 m

First-year ice - thicker than young ice but has less than one year growth, from 0.3 to 2 m

Multi-year ice - ice that has survived at least one melting season (older than one year)

2.4.1 First-Year Ice

The way the first-year ice forms depends on the sea state, among other factors. This means that ice can be formed in calm or rough waters. When in calm sea conditions first crystals are formed as tiny discs ($d < 2 - 3mm$) with vertical c-axis that are growing outwards laterally. As crystals are growing they become unstable and start to break, creating a mixture of ice crystals called frazil or grease ice. As conditions are calm frazil ice crystals will freeze together in order to form a continuous thin sheet of thin nilas ice. As soon as nilas ice is formed, ice growth changes directions and now water molecules freeze on to the bottom of the existing ice sheet. This process is called congelation growth, which yields young ice and subsequently first-year ice.

On the other hand if sea conditions are rough, such as Greenland or Bering Seas where strong sea currents and waves will form a very dense suspension of frazil ice rather than allow for nilas ice to form. This dense mixture of ice crystals and sea water will go through cycles of compression which will in the end create small coherent discs of slush that become more solid as the freeze continues. At the end those discs become known as pancake ice, see Figure 2.4. Size of the pancake ice is relatively small at the edge of

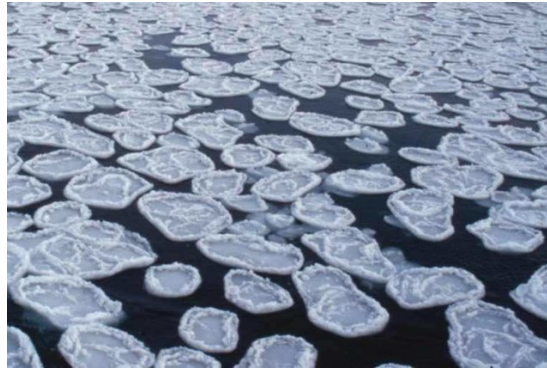


Figure 2.4: Pancake ice in the Barents Sea at the onset of winter. The size of the largest pancakes shown in this figure is roughly one meter, Notz (2005)

the ice field, but grows in diameter and thickness as the distance from the edge becomes larger. At certain point pancake ice will reach 3-5 m in diameter and up to 0.7 m in thickness. At larger distances from the edge of the ice field where sea becomes calmer, individual pancakes will start to coalesce together and form large ice floes that will at the end become continuous sheet of first-year ice known as consolidated pancake ice. Major difference between ice formed in calm and rough conditions is that consolidated pancake ice has very rough and jagged surfaces (bottom and top) compared to relatively smooth surface of first year ice formed in calm conditions. Schematic representation of a cross-section of first-year sea ice can be seen in Figure 2.5. Primary ice in Figure 2.5 is created when continuous sheet of nilas ice is formed and individual ice crystals that are in contact with sea water start to grow downwards. It was mentioned before that individual ice crystals have vertical c-axis when going from frazil to nilas ice, but now freezing process is easier for crystals with horizontal c-axes, which grow at the expense of other ice crystals. If growth of ice crystals is highly influenced by variable sea currents there will be no predominant orientation of c-axis in horizontal plane. Next layer, transition zone, serves as transition to secondary ice composed of long vertical columnar crystals. Columnar structure is a key identifier for ice that grows thermodynamically by freezing onto an ice bottom, which is so called congelation ice. Inside secondary ice brine drainage channels are visible, which are created as the ice-water interface - skeleton layer advances downwards and brine is rejected from the

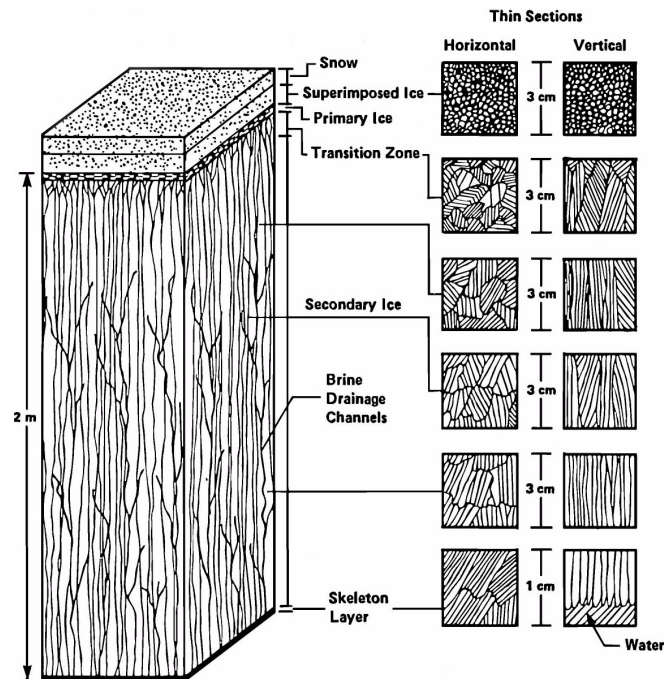


Figure 2.5: Schematic representation of a cross-section of first-year sea ice, Sand (2008)

growing ice sheet. On top of the primary ice there can be a layer of superimposed ice that forms by flooding through ice cracks, refreezing of melted ice or compacting and forming ice from fallen snow.

2.4.2 Multi-Year Ice

Multi-year ice is normally defined as ice which has survived one or more summer seasons of partial melt. Growth will continue until the ice thickness reaches a critical point of about 3 metres. At this point thickness will oscillate throughout annual cycles as the summer melt matches winter growth (Wadhams (2003)). Formation process of multiyear ice is shown on Figure 2.6. Once young/first-year ice is formed it starts to raft caused by dynamic sea conditions - currents and/or waves (Figure 2.6a and 2.6b). First-year ice will grow until reaching a thickness of about 2 m. As summer temperature rises the overlying snow layer begins to melt (in the Arctic this happens during mid-June until early July) and forms a network of meltwater pools on the surface of the ice. As the summer progresses melt pools will become bigger and some will eventually drain

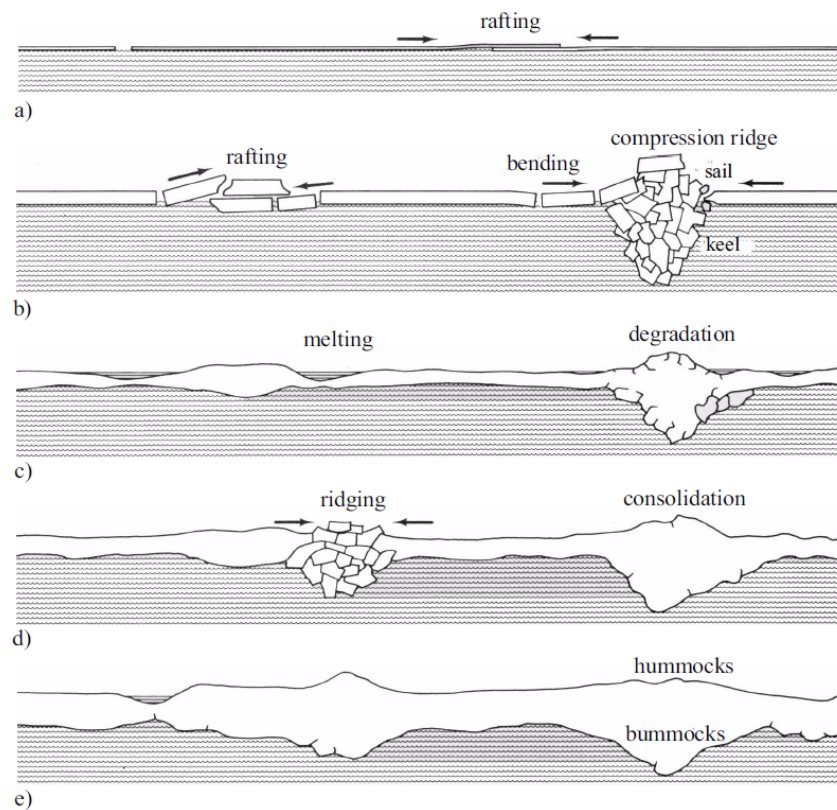


Figure 2.6: Forming of multi-year ice, Sand (2008)

off into the sea. This happens either over the side of ice floes, through existing cracks, or by drilling a thaw hole right through the ice, see Figure 2.6c. What is left over from first-year ice during the second winter will refreeze and consolidate, see Figure 2.6d. This is now called second-year ice. This ice will continue to melt and refreeze over the years and it will become what is known as multi-year ice, see Figure 2.6e.

2.5 Mechanical Properties of Ice

2.5.1 Introduction

Due to complex environmental conditions (temperature, salinity, sea currents, confinement and c-axis orientation among others) sea ice exhibits a wide range of mechanical properties, such as: elasticity, viscoelasticity, viscoplasticity, creep rupture and brittle failure, Schulson (2001). Furthermore sea ice can be described as inhomogeneous,

anisotropic, and nonlinear viscous material. In general properties and behaviour of ice can be classified into two groups - short and long term behaviour. Short term behaviour group encompasses ice that exhibits brittle behaviour and long term group represents ice that displays ductile behaviour. When studying ice properties that are most descriptive of short term behaviour, one needs to examine ice strength characteristics (compressive, tensile, flexural, shear and adhesive) in conjunction with elastic modulus and Poisson's ratio. Long term behaviour is characterized with the same properties but coupled together with time and stress/strain rate. Above mentioned properties are to certain extent dependent on ice grain size, temperature, strain rate, volume/size, c-axis orientation, porosity and brine content and confinement, Sand (2008) and Petrovic (2003).

Temperature - decreasing temperature will increase ice strength in both tension and compression. Note that effects are more prominent in compression.

Strain rate - compressive strength is dependent on strain rate while tensile strength is strain rate insensitive. Sea ice under tensile load exhibits ductile behavior at low strain rates and brittle behavior at intermediate/high strain rates, while under compressive load ice exhibits ductile behavior at low/intermediate strain rates and brittle behavior at high strain rates

Grain size - tensile strength decreases with increasing ice grain size. This relationship is well defined by Hall-Petch definition $\sigma_y = \sigma_0 + kd^m$. Where σ_y is the yield stress, σ_0 is a material constant defining resistance of the lattice to dislocation motion, k is material specific strengthening coefficient and d is the average grain diameter.

Effect of volume/size - with increase in ice specimen size, tensile strength is decreasing.

2.5.2 Elastic Properties

Young's modulus of ice is one of the most studied properties. There are two main types of tests that can be used to assess the Young's modulus; static test such as uniaxial

test under constant strain rate and dynamic test such as high frequency vibrational methods. Problem with measuring Young's modulus of ice by uniaxial test under constant strain rate is that ice shows strong viscoelastic behaviour at temperatures and strain rates usually used for these types of static tests. This means that when load is applied, deformation is measured only after adequate time has passed, which exaggerates viscoelastic effects. Although with good test data, static test should be able to approximate Young's modulus, there is still significant scatter of elastic modulus ranging from 0.3 GPa to 10 GPa. On the other hand a much simpler approach to studying Young's modulus together with its variations with temperature, salinity, air content and so on is to use dynamic tests such as above mentioned high frequency vibrational methods. In vibrational methods displacements are extremely small so that inelastic effects can be neglected. Such methods can involve propagation of waves in small laboratory specimens or large natural ice masses or torsional oscillations of beams. Use of dynamic methods compared to static methods gives less scatter in measured elastic modulus, which is ranging from 6 GPa to 10 GPa, Schulson and Duval (2009) and Mellor (1986).

One of the main parameters that will influence elastic properties of ice is the orientation of ice crystals. According to Schulson and Duval (2009) ice elastic properties can be classified into two main categories; randomly oriented polycrystals and polycrystals with growth textures. As the name says, randomly oriented polycrystal ice has randomly oriented grains that yield isotropic properties. This kind of ice is usually called fine-grained granular ice and can be made by consolidated snow or in extremely wind-blown calm waters. According to Gammon et al. (1983) four sets of elastic moduli from four different ice types (artificial, glacial, sea and lake ice) were weighted and averaged in order to produce a single set of elastic properties, see Table 2.1.

As it was mentioned in previous section crystallographic structure of ice sheets below primary ice (see Figure 2.5) can take different forms depending on conditions, such as sea state, temperature, currents and wind. According to classification by Michel and Ramseier (1971), microstructure of floating ice sheets can be categorized as S1, S2 and S3 class. S1 ice usually develops under calm conditions when small number of

Table 2.1: Polycrystalline Ih ice (Isotropic) average values at -16°C , Gammon et al. (1983)

Property	Units	Value
Young's modulus, E	Nm^{-2}	9.33×10^9
Compressibility, K	N^{-1}m^2	112.4×10^{-12}
Bulk modulus, B	Nm^{-2}	8.90×10^9
Shear modulus, G	Nm^{-2}	3.52×10^9
Poisson's ratio, ν	n/a	0.325

nucleation sites are available, which results in ice with very large crystals whose c-axes are predominantly vertical. S2 ice or columnar-grained ice has randomly oriented c-axes predominantly within the horizontal plane, although as depth increases and crystal become larger, crystals with vertical c-axis will start to edge out those with a more horizontal one. S3 ice defines ice whose c-axes are mostly within horizontal plane of the ice sheet and pointing to a particular direction within that plane (direction can be influenced by strong sea currents), Cole (2001).

Table 2.2: Calculated Young's and shear modulus of homogeneous orthotropic sheets of ice at -16°C , Schulson and Duval (2009)

	Young's modulus, $E \times 10^9\text{N/m}^2$			Shear modulus, $G \times 10^9\text{N/m}^2$		
	E_1	E_2	E_3	G_{12}	G_{13}	G_{23}
S1	9.71	9.71	11.8	3.42	3.01	3.01
S2	9.58^a	9.58^a	9.71	3.61^b	3.21^b	3.21^b
S3	11.8	9.71	9.71	3.01	3.01	3.42

^a VoigtReuss average, Nanthikesan and Sunder (1994)

^b Average shear modulus in $X_1 - X_2$ plane, Nanthikesan and Sunder (1994)

Calculated results for elastic properties of homogeneous orthotropic ice sheet are given in Table 2.2 and Table 2.3. Properties in the tables are given with respect to a rectangular coordinate system. Here X_1 and X_2 are defined in the horizontal plane of the sheet and X_3 is in line with the vertical direction.

For non-saline ice temperature doesn't have a strong effect but for saline ice E will

Table 2.3: Calculated Poisson's ratio of homogeneous orthotropic sheets of ice at -16°C , Schulson and Duval (2009)

	Poisson's ratio					
	ν_{12}	ν_{21}	ν_{13}	ν_{31}	ν_{23}	ν_{32}
S1	0.415	0.415	0.224	0.274	0.224	0.274
S2	0.327 ^a	0.327 ^a	0.344 ^a	0.320 ^a	0.344 ^a	0.320 ^a
S3	0.274	0.224	0.274	0.224	0.415	0.415

^a Estimated from range of Poisson's ratio for monocrystal

increase nonlinearly with decrease in temperature. Although temperature dependence of Young's modulus is nonlinear, for a limited temperature range (0°C to -50°C) value of any fundamental constant such as Young's modulus E can be estimated by the following relationship, Gammon et al. (1983):

$$E(T) = E(T_r)[1 \pm a(T - T_r)] \quad (2.1)$$

where T_r is the reference temperature at which Young modulus was measured, $a = 1.42 \times 10^{-3} \text{K}^{-1}$ and + sign is for compliance and - sign is for stiffness.

Furthermore the value of Young's modulus varies significantly with porosity in sea ice. Porosity of sea ice comes from air bubbles and brine cells. Volume of air bubbles isn't sensitive to temperature change as is the volume of brine cells, which will adjust their volume in order not to change phase as temperature changes. Because most of the studies on mechanical properties of ice take into account only brine volume and not air bubble volume, brine porosity is used when examining most data sets instead of total porosity. In general, Young's modulus of a granular isotropic E_g ice has a linear relation with porosity and can be expressed as follows:

$$E_g = E_0 \left(1 - \frac{\nu_p}{\nu_0} \right) \quad (2.2)$$

where E_0 is elastic modulus of ice without any inclusions (value for Table 2.1 can be

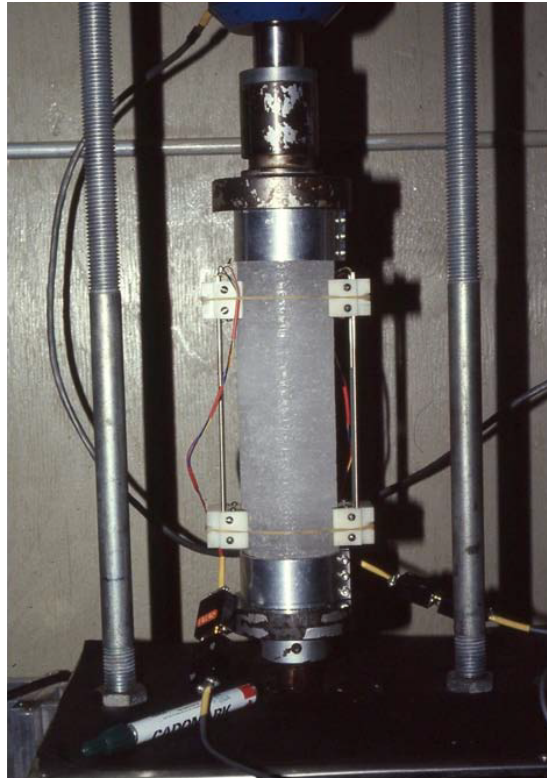


Figure 2.7: Standard setup for uniaxial compression test on sea ice, Timco and Weeks (2010)

used as a reference value), v_p is porosity of ice and v_0 is reference value of the porosity.

2.5.3 Uniaxial Strength

Two most fundamental ice properties are tensile and compressive strength of ice, which are an integral part of complex processes such as ice bending under ships (failure under tension scenario) or ice sheets pushed by currents against each other (failure under compression scenario). In order to better understand such complex processes in ice-structure interactions or interactions between ice floes, an in-depth understanding of uniaxial behaviour of sea ice is needed. For uniaxial compression strength analysis, most common test used is uniaxial unconfined compressive strength test. This test is usually done on cylindrical and prismatic specimens, see Figure 2.7. On the other hand best method for determining uniaxial tensile strength of ice is direct tension testing, as



Figure 2.8: Saline-ice specimen with end caps, Cole et al. (1985)

it yields data that is easier to interpret. For ductile materials uniaxial tensile testing is a straightforward process, but for brittle materials (such as ice under high loading rates) a tiny stress concentration can cause premature failure and give wrong results. In that case special care is needed when preparing specimens, such as mounting special end caps on the specimens which will upon freezing become integral part of the sample, see 2.8.

2.5.3.1 Uniaxial tensile strength

As it was mentioned above, tensile strength is a fundamental property of sea ice and in case of mechanistic studies uniaxial strength becomes the main point of interest. In general sense tensile strength specifies maximum tensile stress that the ice specimen can endure before complete failure. Ice predominantly breaks in tension while under bending, such as, bending under ships or bending when interacting with an offshore

structure, calving of icebergs or buckling when forming pressure ridges, Schulson and Duval (2009).

Polycrystalline granular ice of typical grain size between 1 to 10 mm fails by transgranular cleavage at applied stress between 0.5 MPa and 2 MPa. There are several factors that affect tensile failure, such as temperature, grain size, strain rate and porosity (brine + gas). From all of the mentioned factors the one that has the least impact on tensile failure is the strain rate. This is at least true for quasi-static strain rates that range between 10^{-7} s^{-1} and 10^{-1} s^{-1} where strain rate essentially has no impact on failure. This is however not so true for strain rates in dynamic range where ice seems to display strain-rate hardening. For example Lange and Ahrens (1983) showed that ice at strain rates of $\sim 10^4 \text{ s}^{-1}$ has a tensile strength of $\sim 17 \text{ MPa}$. Strain rate will also affect ductile and brittle failure of ice, this means that granular ice with 1 mm grain size at -10°C transitions from ductile to brittle failure at strain rate of 10^{-7} s^{-1} .

Temperature compared to strain rate has a larger impact on tensile strength. Compared to fresh-water ice, sea water ice displays greater thermal sensitivity. Menge and Jones (1993) have done tensile tests on first-year columnar sea ice over a range of temperatures from -20°C to -3°C and strain rates between 10^{-5} s^{-1} to 10^{-3} s^{-1} . Furthermore, they have compared their results to research done by Dykins (1970) and Kuehn et al. (1990). What was immediately evident is that temperature has a strong influence on a cross-column strength of columnar-grained sea ice, infact the increase was about a factor of four: $\sim 0.2 \text{ MPa}$ at -2.5°C to $\sim 0.9 \text{ MPa}$ at -20°C , see Table 2.4.

As it can be seen from the Table 2.4 first-year ice is extremely dependent on temperature, due to the influence of temperature on porosity/brine pockets volume within salt-water ice. As the temperature decreases, the equilibrium volume fraction of brine decreases until NaClH_2O eutectic is reached at -21.2°C . Dependency on temperature is further reinforced because the brine inclusions in a platelet-like manner are arranged along lines that coincide with the plane of failure.

From all above mentioned factors, grain size is the most important factor when it comes to tensile strength of ice. According to Schulson and Duval (2009) there is a

Table 2.4: The tensile strength of columnar-grained first-year sea ice loaded uniaxially across the columns

Temperature °C	Rate s ⁻¹	Tensile Strength MPa
-20	1.0 ⁻³	0.78 ± 0.14
-20	1.0 ⁻⁵	0.73 ± 0.07
-10	1.0 ⁻³	0.63 ± 0.12
-10	1.0 ⁻⁵	0.56 ± 0.06
-5	1.0 ⁻³	0.47 ± 0.13
-5	1.0 ⁻⁵	0.45 ± 0.06
-3	1.0 ⁻³	0.32 ± 0.06
-3	1.0 ⁻⁵	0.21 ± 0.09

large body of research that shows a strong inverse correlation between grains size d and tensile strength σ_t , which states that increase in d will result in decrease in σ_t . Some of the studies that prove this relationship can be found in Hawkes and Mellor (1972) and Schulson et al. (1984). In the study by Schulson et al. (1984) standardized measurements on cylindrical-shaped specimens of granular fresh-water ice were done at -5°C , -10°C and -20°C with the grain size ranging from 0.8 to 9.4 mm. This paper first proved that by increasing the grain size, tensile strength will decrease. Secondly it also established that there is a critical grain size marks ductile-to-brittle transition. Critical grain size was measured between 1.5 and 1.9 mm. For grains larger than critical size tensile failure are governed by crack nucleation and for grains smaller then critical size tensile failure are controlled by crack propagation. This means that critical grain size defines the point where crack nucleation stress becomes equal to crack propagation stress. According to Schulson et al. (1984) for lower strain rates (10^{-7} s^{-1}) uniaxial tensile strength can be expressed as Orwan form:

$$\sigma_t = Kd^{-\frac{1}{2}} \quad (2.3)$$

where K is measure of fracture toughness of granular ice and d is the grain size. At higher strain rates (10^{-3} s^{-1}) the function takes the Hall-Petch form:

$$\sigma_t = \sigma_0 + k_t d^{-\frac{1}{2}} \quad (2.4)$$

where σ_0 and k_t are measures of frictional resistance to dislocation glide and the effectiveness with which grain boundaries impede slip and d is grain size as in Equation 2.3. As it was stated by Schulson and Duval (2009) Equations 2.3 and 2.4 need to be used with certain amount of caution. Although these equations should be general, parameters are probably still affected by the type of ice that was used when obtaining the values - randomly orientated granular fresh-water ice without cracks. The concept of critical grain size d_c suggests that for finely grained ice ($d < d_c$) nucleated cracks are more stable so that ice can support stress greater than stress needed to nucleate the first crack. For more coarsely grained ice ($d > d_c$) nucleated cracks are unstable so fracture starts immediately. The critical grain size d_c is obtained by equating Equations 2.3 and 2.4 as follows:

$$d_c = \left(\frac{K - k_t}{\sigma_0} \right)^2 \quad (2.5)$$

Temperature and porosity are mutually connected parameters, where change in temperature will affect the equilibrium volume fraction of brine. Furthermore, Menge and Jones (1993) showed that there is a nonlinear decrease in tensile strength with increasing porosity (brine+gas). Figure 2.9 shows tensile strength as a function of porosity for horizontally loaded first year sea ice. By neglecting relatively small effect of strain rate, following function for tensile strength can be extrapolated;

$$\sigma_t = 4.278v_t^{-0.6455} \quad (2.6)$$

where v_t is function of total porosity. In general salt-water ice is much more complicated material than fresh-water ice. This is because of its microstructure, since c-axis orientation and grain size can be controlled, but other factors such as porosity are much more difficult to regulate. With this in mind alternative conclusions could be made, such as size and spacing of inclusion could be more important than grain size. Also crack nucleation could be irrelevant for limiting tensile strength, because brine pockets may act like cracks. Because of this, providing definite conclusions from sea-water ice testing is much more difficult than for fresh-water ice. This means that clear conclusions are difficult to make.

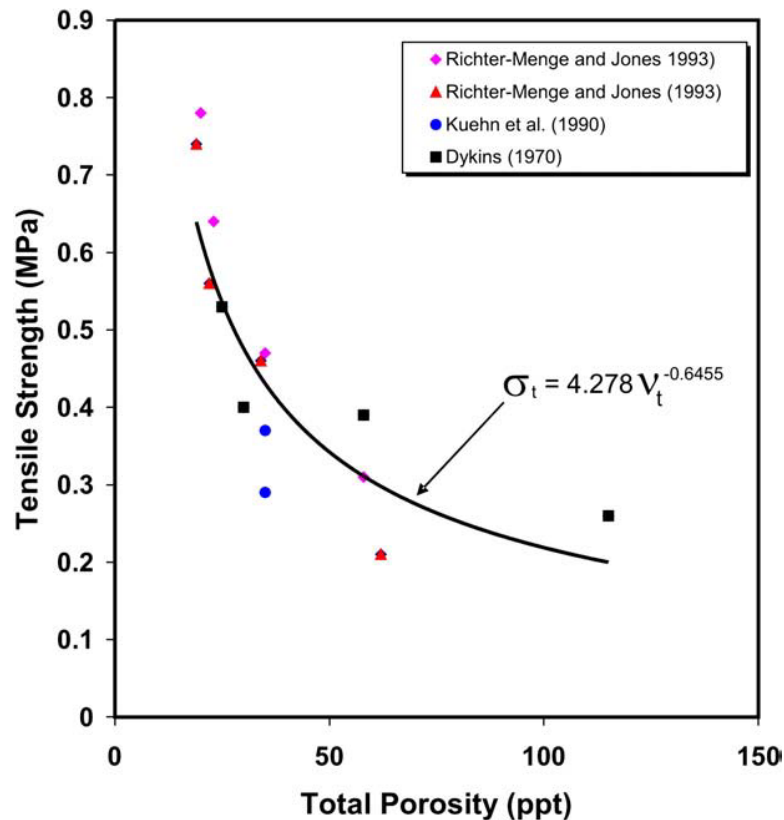


Figure 2.9: Tensile strength as a function of porosity for horizontally loaded first year sea ice, Timco and Weeks (2010)

2.5.3.2 Uniaxial compressive strength

The compressive strength of sea ice is also one of the fundamental properties. Studies of both small and large-scale sea ice failure have concluded that ice often fails in compression. There are variety of scenarios where compressive failure happens, such as ice floe impacting an offshore structure or formation of large pressure ridges. Because of its importance, a large body of in-depth studies were done. As it was mentioned before, one of the most common test used in the studies was uniaxial unconfined compressive strength test as it is relatively easy to set up and the data is very easy to interpret. As with tensile strength, compressive strength of ice is also affected to a certain degree by a variety of factors. These factors can be intrinsic, such as temperature, salinity, density, ice type, crystal size and orientation or part of test setup, rate of loading, confinement conditions, sample size, stiffness of the test machine, sample preparation techniques

and loading direction - which is measured relative to the orientation of columnar grains (Timco and Weeks (2010)).

When subjected to the compressive load, ice will exhibit two kinds of inelastic behaviour - ductile and brittle behaviour, see Figure 2.10. Transition from ductile to brittle and vice versa occurs gradually rather than abruptly. Main reason behind gradual transition between ductile and brittle behaviour is because inelastic deformation is defined by a mixture of creep by dislocation slip and cracking. Moving closer to transition point $\dot{\epsilon}_{D/B}$, ratio of creep and cracking is changing from dominant creep rupture in ductile range to crack nucleation and crack propagation in brittle range.

Ductile behaviour is characterized by strain-rate hardening owned to grain boundary sliding and dislocation glide and climb within grains - secondary creep, see Figure 2.10. According to Michel (1978b) at lower strain rates stress-strain curve will level off and at higher strain rates it will reach maximum stress instantly followed by strain softening. Strain softening is due to internal cracking and dynamic recrystallization, which is characterized by ice that is initially transparent and then becoming milky-white in appearance due to non-propagating, grain-sized microcracks. Furthermore, plastic strain in excess of 0.1 can be reached without macroscopic collapse. Strain-rate hardening can be expressed by the following expression (Schulson and Duval (2009)):

$$\sigma \sim \dot{\epsilon}^{1/n} \quad \text{where } n \sim 3 \quad (2.7)$$

In addition, ductile behaviour is also characterized by thermal softening (maximum stress increases with decreasing temperature) and it is sensitive to salinity and porosity (maximum stress decreases with increasing salinity/porosity). On the other hand stress values exhibit little dependence on grain size. Compared to ductile behaviour, brittle behaviour is defined by a stress-strain curve that rises linearly and terminates suddenly after reaching inelastic strain of < 0.003 due to onset of a mechanical instability. Ice exhibits brittle behaviour at rates above $\approx 10^{-4} - 10^{-3} \text{ s}^{-1}$. In brittle regime ice will fail differently depending on level of confinement. If ice is unconfined it will fail via multiple splitting along the direction of loading and by faulting or spalling if it is confined. The

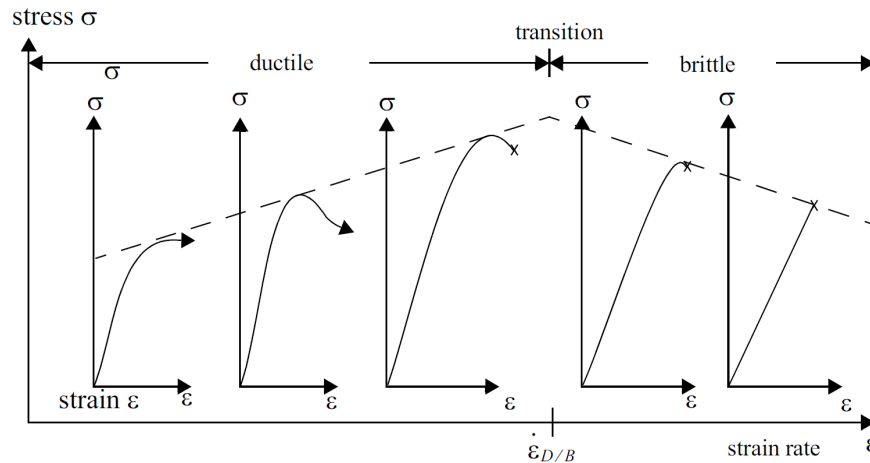


Figure 2.10: Transition between ductile and brittle failure of ice under compressive load, Sand (2008)

terminal stress or failure strength is a function of temperature, strain rate, grain size and confinement. Terminal stress will decrease with increasing temperature by a factor of ~ 2.5 with temperature going from -50°C to -10°C , with increasing strain rate by a factor of ~ 1.4 with strain rate going from 10^{-3} to 10^{-1} s^{-1} and with increasing grain size by a factor of ~ 3 with grain sizes from 1 to 10 mm. When compared to uniaxial tensile strength of ice, brittle compressive strength seems to be unaffected by the salinity/porosity, at least at -10°C .

Brittle compressive failure is a multi-step process that is controlled by crack nucleation and crack propagation. It begins with crack nucleation at grain boundaries due to dislocation pile-up. As the stress increases, other cracks nucleated throughout the body. As soon as the shear traction induced by the far-field stress is great enough to overcome frictional resistance, most favorably oriented cracks begin to slide which leads to formation of wing cracks. With further load increase wing cracks start to grow mostly trans-granularly and start to interact with each other. Final failure generally occurs as large number of cracks coalesce and wing cracks propagate all the way to the free edges of the specimen and form a longitudinal split.

As it can be seen from the Figure 2.10 that compressive strength reaches its maximum value at the ductile-to-brittle transition point. This happens as a consequence of both strain-rate hardening in the ductile regime and strain-rate softening during the

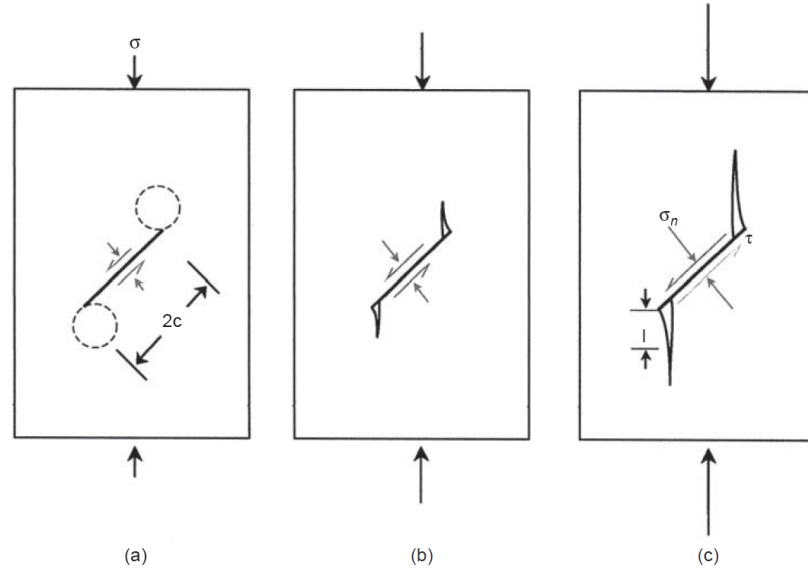


Figure 2.11: Beginning stages in the development of wing cracks under compressive loading. (a) Initial low stress develops tensile zones around the tips of the inclined cracks. (b) By increasing stress wing cracks initiate at the inclined crack tips. This is due to frictional sliding across inclined crack. (c) At even bigger stress, the wing cracks propagate and separate their faces due to further sliding, Schulson and Duval (2009)

brittle regime.

Examining 283 compressive strength test results, Timco and Frederking (1990, 1991) have developed a model for determining strength of sea ice sheets. Examined tests included the following variables: temperature, ice salinity, bulk ice density, grain structure, loading direction, number of tests and test results. From there they have derived several equations as a function of grain structure. Uniaxial compressive strength for horizontally loaded columnar ice was derived as following:

$$\sigma_c = 37(\dot{\epsilon})^{0.22} \left[1 - \sqrt{\frac{v_T}{270}} \right] \quad (2.8)$$

for vertically loaded columnar ice as:

$$\sigma_c = 160(\dot{\epsilon})^{0.22} \left[1 - \sqrt{\frac{v_T}{200}} \right] \quad (2.9)$$

and for granular ice as:

$$\sigma_c = 49(\dot{\epsilon})^{0.22} \left[1 - \sqrt{\frac{v_T}{280}} \right] \quad (2.10)$$

where v_T is the total porosity (sum of brine and air porosity). Above derived equations are relevant for strain rates between 10^{-7} s^{-1} to $2 \times 10^{-4} \text{ s}^{-1}$ and give an explicit correlation between grain type, loading direction, loading strain rate and total porosity, and implicitly to ice salinity, temperature and density.

2.5.4 Shear strength

When analyzing ice-structure interactions, ice is usually subjected to a complex stress condition. This means that ice cover is often under biaxial stress conditions composed from tensile and compressive stress or shear stress. Thus, together with uniaxial compressive and tensile strength of ice, shear strength is also a fundamental property of ice. Compared to uniaxial tensile and compressive tests, shear testing is more difficult to setup/execute and interpret due to generating stress fields that cannot be quantified in a simple manner. When conducting shear testing one is assuming that the only stress condition generated is uniform shear stress, but in reality additional normal stress is also generated on the plane of failure, whose value is impossible to be determined. To this day primary methods for determining shear strength are torsion, direct shear and punching test.

Throughout the literature there have been relatively few measurements of the shear strength and the ones that exist show shear strength as a function of temperature, salinity, density, and for columnar sea ice the shear strength is additionally dependent on loading direction measured relative to the columnar grains. Although tests have shown that there is a direct correlation between strain rate and compressive strength, the same correlation can't be established with certainty for shear strength. Butkovich (1956) used double shear device on cylindrical specimens made out of first year ice with approximately 6% salinity. Specimens were loaded perpendicular to the long axis of the columnar grains. Obtained average shear strength was 1.6 MPa for temperatures between -5°C to -7°C and 2.3 MPa for temperatures between -10°C to -13°C . On the

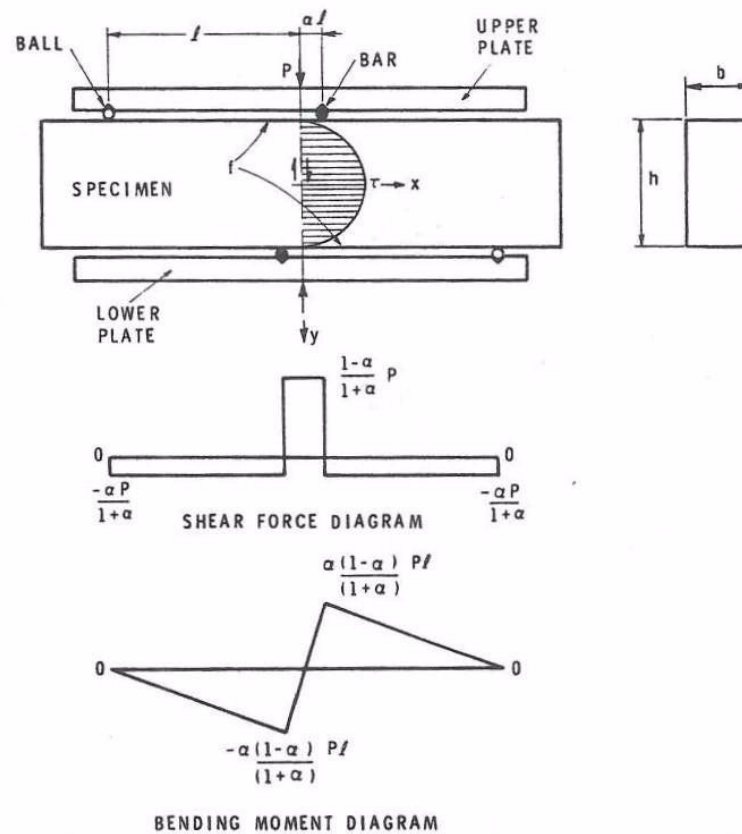


Figure 2.12: Asymmetric four-point loading apparatus together with shear force and bending moment diagrams, Frederking and Timco (1986)

other hand Paige and Lee (1967) and Dykins (1971) carried single shear experiments on similar sized cylindrical specimens but with loading oriented in such a way that failure plane developed along specimens length parallel to the columnar grains. Furthermore, a relief hole was drilled in the central section in order to reduce confinement effects. Paige and Lee (1967) used specimens from natural sea ice and obtained shear strength between 0.5 MPa and 1.2 MPa with a significant dependence on brine volume. While Dykins (1971) used artificially-grown columnar-grained saline ice and obtained shear strength between 0.1 MPa and 0.25 MPa. In all three cases experiments show large discrepancies despite using ice with similar salinity and temperature. This can be attributed to the normal stresses that develops on the failure plane as a consequence of single and double shear test devices.

Table 2.5: Shear strength of columnar-grained sea ice. Grain orientation is measured relative to shear plane Frederking and Timco (1986).

Grain orientation	Temperature °C	Density kg/m ³	Total porosity ppt	Shear Strength MPa
vertical	-2	830	136	0.715
horizontal	-2	830	136	0.560
vertical	-12	850	83	0.645
horizontal	-12	850	83	0.760

With this in mind Frederking and Timco (1986, 1984) have carried out shear tests by utilizing an asymmetric 4-point bending system on sea ice specimens as illustrated in Figure 2.12. By using asymmetric 4-point bending system loads were applied in a manner so that a high shear stress state is created in the mid-section of the beam. For the granular ice they have estimated an average shear strength of $0.55 \text{ MPa} \pm 0.12 \text{ MPa}$ with test temperatures at $-13^\circ\text{C} \pm 2^\circ\text{C}$ and salinity of $4.2\% \pm 0.5$. Frederking and Timco (1986) measured the shear strength of columnar sea ice for two different specimen orientations. First test case was horizontal shear where the shear plane is perpendicular to the long axis of the columnar grains. Second test case was vertical shear where the shear plane is parallel to long axis of the columnar grains. Results for the shear strength of columnar grained sea ice is summarized in Table 2.5. Furthermore, Frederking and Timco (1986) used Paige and Lee (1967) results in combination with their own results in order to extrapolate vertical shear strength σ_s as a function of total porosity v_T as follows:

$$\sigma_s = 1.5 \left(1 - \sqrt{\frac{v_T}{390}} \right) \quad (2.11)$$

In general, shear strength isn't commonly used as it is governed by tensile strength of sea ice. Furthermore, most engineering scenarios include higher loading rates with compressive strength being higher than the tensile strength. Hence, ice subjected to shear condition will fail in tension rather than in shear.

2.5.5 Fracture Toughness

Basic idea behind fracture mechanics is to establish a quantitative description of the transition from a virgin material into a fractured and fragmented material via crack growth. Most basic fracture mechanism occurs in brittle solids, with development of critical stress at which the crack becomes unstable and propagates. Crack tip stress field σ_{ij} can be described as a function of the stress intensity factor K - ratio of true stress at the crack tip and the average applied stress:

$$\sigma_{ij} = f(K, r, \theta) = K f_{ij}(r, \theta) \quad (2.12)$$

where r and θ are polar coordinates with the origin at the tip. By increasing applied stress, value of K will increase until it reaches critical value K_c at which fracture occurs. Depending on the loading conditions there can be three distinct critical stress intensity factors that corresponds to three modes of crack propagation, see Figure 2.14. K_{Ic} refers to stress normal to the crack plane condition (mode I), K_{IIc} where stress is applied in such manner so that the crack faces move over each other perpendicular to the crack front (mode II) and K_{IIIc} refers to stress applied parallel to the crack front (mode III). When considering crack propagation it is also beneficial to consider energy release rate G - rate of energy release per unit length of crack advance. Under plane-strain conditions K_{Ic}^2 and G are related as:

$$G = K_{Ic}^2 \left(\frac{1 - \nu^2}{E} \right) \quad (2.13)$$

or in plane-stress as:

$$G = \frac{K_{Ic}^2}{E} \quad (2.14)$$

Griffith (1921) derived critical stress required for crack propagation by equating energy release rate G with rate of increase for specific surface energy γ_s as:

$$\sigma_c = \left(\frac{2E\gamma_s}{\pi c} \right)^{1/2} \quad (2.15)$$

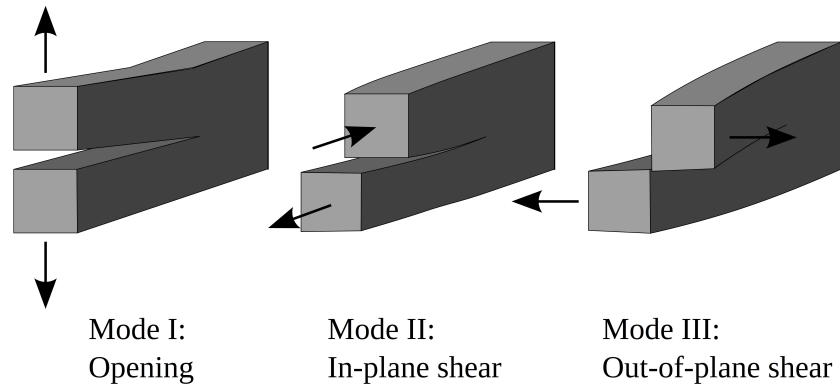


Figure 2.13: Three modes of crack propagation

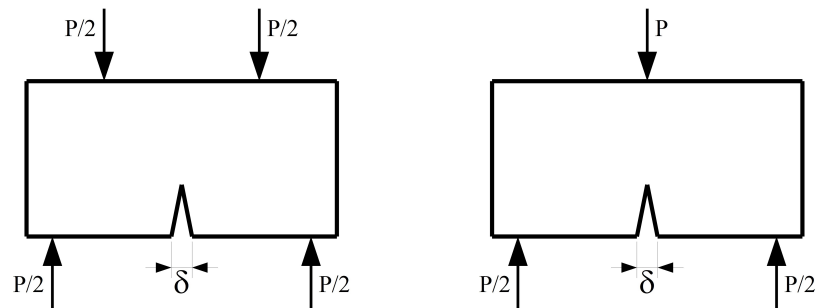


Figure 2.14: Left image: Four-point-bend configuration and Right image: Three-point-bend configuration

or for plane-stress

$$\sigma_c = \left(\frac{2E\gamma_s}{(1-\nu^2)\pi c} \right)^{1/2} \quad (2.16)$$

where c is crack length. Fracture toughness of ice is usually measured by using 4-point or 3-point beam configuration, see Figure 2.14. Beams will have a notch cut in the centre that has a very small tip (usually created by using a razor). Length of the notch is approximately one-third of beam's thickness. In order to calculate the fracture toughness two values are usually used - load at failure or COD (crack opening displacement) of the notch, Timco and Weeks (2010).

Fracture toughness of ice mostly depends on the loading rate and crystallographic structure of ice, with less variation due to temperature, grain size and porosity. In

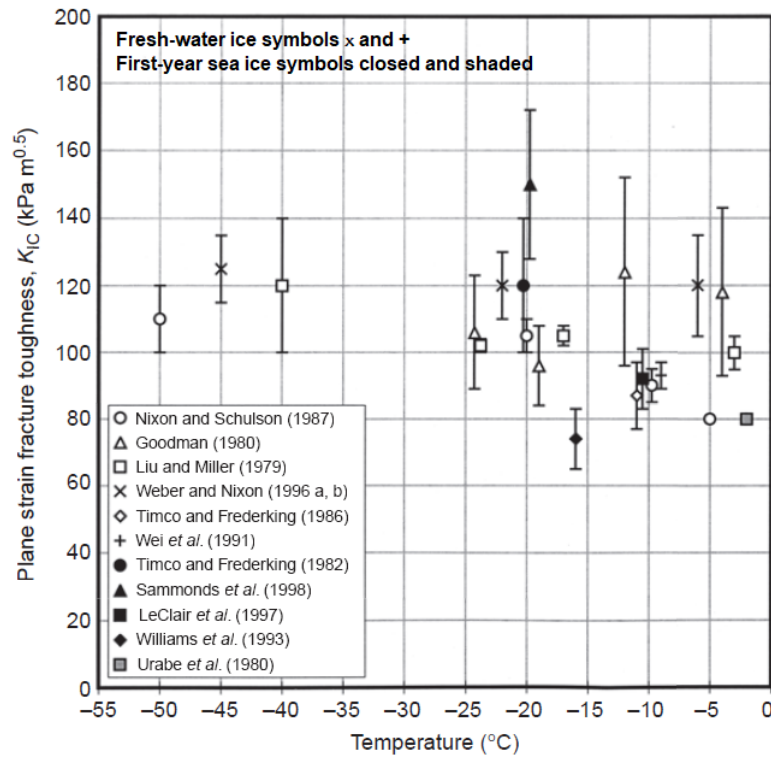


Figure 2.15: Plane-strain fracture toughness vs. temperature, Schulson and Duval (2009)

Figure 2.15 plane-strain fracture toughness vs. temperature is shown from multiple studies and one can see that most of the data is clustered around $K_{Ic} = 100 \text{ kPa m}^{1/2}$ with considerable scattering. With this in mind it is hard to detect clear effects of temperature on fracture toughness.

Nixon and Schulson (1988) found that increase in grain size will decrease the fracture toughness, through the relationship:

$$K_{Ic} = K_{Io} + \phi d^{-1/2} \quad (2.17)$$

where d is the grain size in mm, $K_{Io} = 58.3 \text{ kPa m}^{1/2}$ and $\phi = 42.4 \text{ kPa m}^{1/2}$. In general porosity will reduce the stiffness of ice, which was observed by Rist *et al.* (1999). Furthermore, Rist *et al.* (2002) found the linear relationship between porosity

p and fracture toughness if one assumes $p < 0.4$ as:

$$K_{Ic}(p) = K_{Ic}(1 - p) \quad (2.18)$$

Chapter 3

Peridynamic Theory

3.1 Introduction

The peridynamic theory (PD) was introduced by Silling (2000). PD theory deals with the motion of a material point that interacts with all points within its range (see Figure 3.1). Peridynamic theory uses displacements rather than displacement derivatives in its formulation. Hence, as opposed to classical continuum mechanics formulation which uses partial differential equations, peridynamic theory is based on integral equations which are valid at fracture surfaces. It is basically the re-formulation of classical continuum mechanics (CCM) equations using integro-differential equations, in which derivatives only come into picture with time derivatives of displacements. Both theories assume that a domain, V , can be discretized into many infinitesimal volumes, i.e. material points and their interactions. In CCM, the material point at position \mathbf{x} only interacts with its nearest neighbors whereas in PD theory it can interact with material points \mathbf{x}' which are not only limited to the nearest neighbors of \mathbf{x} . The former local interactions are in the form of traction vectors, \mathbf{T} as opposed to the nonlocal force densities, \mathbf{t} and \mathbf{t}' in PD theory. In CCM, the traction vectors are expressed in the form of stress tensor, $\boldsymbol{\sigma}$ and its equation of motion can be expressed as

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) = \nabla \cdot \boldsymbol{\sigma} + \mathbf{b}(\mathbf{x}, t) \quad (3.1)$$

where $\nabla \cdot$ indicates a divergence, \mathbf{u} and \mathbf{b} are the displacement and body load vectors, respectively. Moreover, \ddot{u} stands for acceleration vector, ρ and t denote the density of the material and time and, respectively. On the other hand, Peridynamic theory has the following form of equation of motion

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{H_x} (\mathbf{t}(\mathbf{x}' - \mathbf{x}, \mathbf{u}' - \mathbf{u}) - \mathbf{t}'(\mathbf{x} - \mathbf{x}', \mathbf{u} - \mathbf{u}'))dV_{x'} + \mathbf{b}(\mathbf{x}, t) \quad (3.2)$$

where integration region or horizon H_x includes all interactions between the main material point \mathbf{x} and its family members \mathbf{x}' . When comparing first term on RHS of Equation 3.1 and Equation 3.2 we can see that divergence of a stress tensor $\boldsymbol{\sigma}$ which is a vector valued function is mirrored by an integral over a force density vector \mathbf{t} . The range of interactions, generally called as bonds, are limited with the horizon radius, δ around the material point \mathbf{x} . The force density, \mathbf{t} arises from the relative displacements of all material points, \mathbf{x}' with respect to material point \mathbf{x} within the horizon of the material point \mathbf{x} . In the original form of PD theory, which is named as bond-based PD theory, it is assumed that the force densities have the same magnitude and this assumption causes a restriction on material constants which lead to one material constant instead of two constants for isotropic materials. PD theory has attracted attention of many researchers from all over the world. The reason for this is mainly the capability of PD theory for modeling discontinuities over a domain. Many researchers succeeded to solve many challenging and diverse problems involving discontinuities using peridynamics especially after the publication of successive papers by Silling and Askari (2005) and Macek and Silling (2007). The former concentrated on numerical implementation part of the bond-based PD theory and the latter is its demonstration by using commercial FE software, Abaqus.

Following the introduction of most general forms of PD theory by Silling et al. (2007), which are so called ordinary and non-ordinary state based theories, the interest on PD theory has been dramatically increased, since the limitation on material constants was removed and it paved the way of modeling more challenging problems such as plastic deformation Madenci and Oterkus (2016). Moreover, the application of

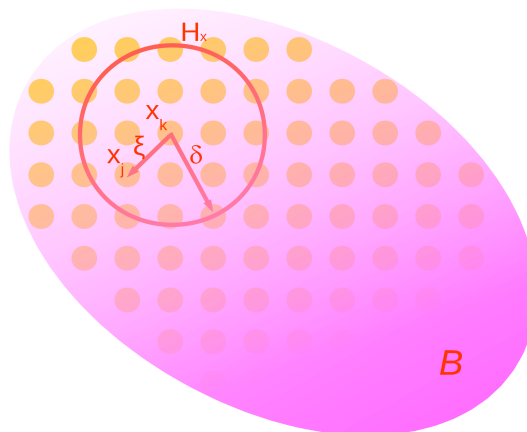


Figure 3.1: The material point \mathbf{x} interacts with other material points inside its horizon H_x

peridynamics has also been extended to other fields such as thermal diffusion Oterkus et al. (2014), electric flow Oterkus et al. (2013) and hydrogen diffusion De Meo et al. (2016). The book by Madenci and Oterkus (2014) presents the numerical implementation areas of PD theory including complex laminated composite materials. Moreover, a recent book by Madenci et al. (2019) brings a new aspect on solving many formidable differential equations by using PD differential operator.

The Peridynamic equation of motion is usually solved by using meshless discretization methods as explained in the Section 4.4. Furthermore, Emmrich and Weckner (2007) compared all of the solution methods including the finite element (FE) method for a one-dimensional PD problem which does not contain any discontinuities. In that case, the FE method has the best accuracy amongst the others. However, it requires more computational time to solve the matrix equations. Besides, the meshless methods are the most convenient ones for PD problems with discontinuities. The solution does not involve any jump terms as in the CCM theory because the governing equation of PD theory is derivative-free. For this reason, most of the researchers (Madenci and Oterkus (2014); Macek and Silling (2007)) prefer meshless midpoint rule for numerical implementation of PD theory. It is simple and easily applicable when discontinuities exist in the body. Therefore, the peridynamic equation of motion given in Eq. (3.2)

can be written in a discrete form as

$$\rho(\mathbf{x}_i)\ddot{\mathbf{u}}(\mathbf{x}_i, t) = \sum_{j=1}^{N_i} (\mathbf{t}(\mathbf{x}_j - \mathbf{x}_i, \mathbf{u}_j - \mathbf{u}_i) - \mathbf{t}'(\mathbf{x}_i - \mathbf{x}_j, \mathbf{u}_i - \mathbf{u}_j))V_j + \mathbf{b}(\mathbf{x}_i, t) \quad (3.3)$$

where N_i is the number of material points inside the horizon of the material point at \mathbf{x}_i and V_j is the volume of the material point at \mathbf{x}_j .

The most general in-house PD code mainly contains the following steps:

1st step - Constructing material points: The body is composed of many small finite volumes and the center of each volume is represented by a material point located at the center of the volume. In this step, the material points are created while determining their locations in a global coordinate system.

2nd step - Family member search: Family member points, which reside inside the horizon of each main material point, are determined and the family member array is created.

3rd step - Surface correction: The horizon is usually truncated near the boundaries of a surface and this results in reduction of material point stiffness. Hence, the stiffness of material points near the free surfaces is corrected (see Chapter 4 of ref. Madenci and Oterkus (2014)). For more information about different surface correction approaches available in the literature please see Le and Bobaru (2018).

4th step - Time integration: The PD equations are solved dynamically or statically.

3.2 Basics of Peridynamic Theory

In peridynamic theory continuum is defined by infinitely many material points (particles), where each particle interacts with the surrounding points. In undeformed state

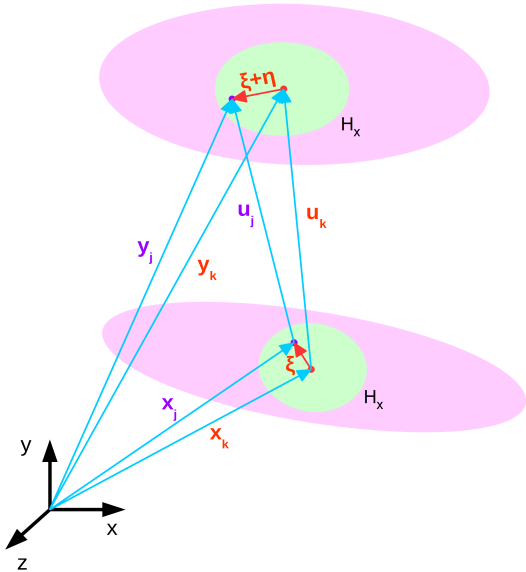


Figure 3.2: The material point \mathbf{x} interacts with other material points inside its horizon H_x

each particle is defined with respect to the Cartesian system by: initial position \mathbf{x}_k ($k = 1, 2, \dots$), incremental volume V_k , and mass density ρ . The deformed state of the material point is described by position vector \mathbf{y}_k that is the result of external loading. The motion of a material point is the result of forces acting upon the material point. External loading can be in the form of the body load \mathbf{b}_k , displacement \mathbf{u}_k , or velocity \mathbf{v}_k . According to Silling (2000), motion of the material point x_k , is calculated as an collective contribution from other points within its influence zone. Influence zone or the horizon can be infinitely large, but is presumed that effects vanish beyond certain range. Although horizon is a very important parameter, research on how to choose this parameter has been rather limited and mainly depends on suggestions made in the influential paper written by Silling and Askari (2005). They suggested to use a horizon size equivalent to three times of the grid spacing between material points based on the experiences of these researchers for their simulations. Influence range or horizon radius is defined by δ and influence region or horizon is denoted by H_x (see Figure 3.2).

As it can be seen from Figure 3.2, the material point \mathbf{x}_k is influenced by the collective deformation of all material points inside its horizon H_x . The same can be stated for

the material point \mathbf{x}_j . The deformed position \mathbf{y}_k and \mathbf{y}_j is acquired by applying their displacements \mathbf{u}_k and \mathbf{u}_j , respectively, to the initial position \mathbf{x}_k and \mathbf{x}_j . The stretch between material points \mathbf{x}_k and \mathbf{x}_j is defined as:

$$s_{kj} = \frac{|\boldsymbol{\eta} + \boldsymbol{\xi}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \quad (3.4)$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are the relative position in reference configuration and the relative displacement. They can be expressed as $\boldsymbol{\xi} = \mathbf{x}_j - \mathbf{x}_k$ and $\boldsymbol{\eta} = \mathbf{u}_j - \mathbf{u}_k$. In the deformed configuration each bond between material points will experience a gain in energy, which is called micropotential. Scalar-valued micropotential w_{kj} depends on the material properties and the stretch between point \mathbf{x}_k and other material points in its family (denoted by $m = 1..\infty$), while w_{jk} depends on the material properties and the stretch between point \mathbf{x}_j and other material points in its family (denoted by $n = 1..\infty$). This would than mean that $w_{kj} \neq w_{jk}$.

$$\begin{aligned} w_{kj} &= w_{kj}(y_1 - y_k, y_2 - y_k, \dots, y_{m-1} - y_k, y_m - y_k) \quad \text{where } m = 1..\infty \\ w_{jk} &= w_{jk}(y_1 - y_j, y_2 - y_j, \dots, y_{n-1} - y_j, y_n - y_j) \quad \text{where } n = 1..\infty \end{aligned} \quad (3.5)$$

where \mathbf{y}_k is the deformed position vector of point \mathbf{x}_k and \mathbf{y}_m is the deformed position vector of the material point that is inside of the horizon of \mathbf{x}_k . Similarly, \mathbf{y}_j is the deformed position vector of point \mathbf{x}_j and \mathbf{y}_n is the deformed position vector of the material point that is inside of the horizon of \mathbf{x}_j .

The strain energy density, W_k of material point \mathbf{x}_k can be expressed as a summation of micropotentials, w_{kj} , arising from the interaction of material point \mathbf{x}_k and other material points, \mathbf{x}_j , within its horizon as

$$W_k = \frac{1}{2} \sum_{j=1}^{\infty} \frac{1}{2} (w_{kj}(\mathbf{y}_m - \mathbf{y}_k) + w_{jk}(\mathbf{y}_n - \mathbf{y}_j)) V_j \quad (3.6)$$

where $w_{kj} = 0$ for $k = j$. According to Silling and Askari (2005), factor $\frac{1}{2}$ before the summation comes from the fact that each endpoint of a bond owns only half the energy in the bond. Furthermore the factor $\frac{1}{2}$ under the summation comes from the fact that

micropotential of a bond can be assumed as the average of w_{kj} and w_{jk} .

Definition of the strain energy density for a point \mathbf{x}_k was a first step towards defining equation of motion in peridynamic terms. Equation of motion of \mathbf{x}_k can be derived through application of principle of virtual work. This approach can be summarized by Hamilton's principle (see Equation 3.7) which states that the development in time for a mechanical system is such that the integral of the difference between the kinetic and the potential energy is stationary. The difference between kinetic and the potential energy is defined as Lagrangian (L).

$$\delta \int_{t_0}^{t_1} L dt = 0 \quad (3.7)$$

Hamilton's principle is satisfied by Lagrange's equations of the second kind or the Euler-Lagrange equations of motion for material point \mathbf{x}_k

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{u}}_k} \right) - \frac{\partial L}{\partial \mathbf{u}_k} = 0 \quad (3.8)$$

As it was mentioned above Lagrangian is defined as difference between total potential and kinetic energy.

$$L = T - U \quad (3.9)$$

In peridynamic case body's total potential and kinetic energy are defined by sum of kinetic and potential energies of all material points. Thus, total potential energy can be defined as:

$$U = \sum_{k=1}^{\infty} W_k V_k - \sum_{k=1}^{\infty} (\mathbf{b}_k \mathbf{u}_k) V_k \quad (3.10)$$

and total kinetic energy as

$$T = \sum_{k=1}^{\infty} \frac{1}{2} \rho_k \dot{\mathbf{u}}_k \dot{\mathbf{u}}_k V_k \quad (3.11)$$

Strain energy density, W_k , in Equation 3.10 can be substituted by strain energy density of material point \mathbf{x}_k for Equation 3.6 and thus total potential energy can be

rewritten in more appropriate form

$$U = \sum_{k=1}^{\infty} \frac{1}{2} \left\{ \sum_{j=1}^{\infty} \frac{1}{2} [w_{kj}(\mathbf{y}_m - \mathbf{y}_k) + w_{jk}(\mathbf{y}_n - \mathbf{y}_j)] V_j - (\mathbf{b}_k \mathbf{u}_k) \right\} V_k \quad (3.12)$$

Once total potential and kinetic energy of the body is defined, Lagrangian can be obtained by substituting Equation 3.10 and Equation 3.11 into Equation 3.9.

$$L = \sum_{k=1}^{\infty} \frac{1}{2} \rho_k \dot{\mathbf{u}}_k \dot{\mathbf{u}}_k V_k - \sum_{k=1}^{\infty} \frac{1}{2} \left\{ \sum_{j=1}^{\infty} \frac{1}{2} [w_{kj}(\mathbf{y}_m - \mathbf{y}_k) + w_{jk}(\mathbf{y}_n - \mathbf{y}_j)] V_j - (\mathbf{b}_k \mathbf{u}_k) \right\} V_k \quad (3.13)$$

In order to simplify the solution of Euler-Lagrange equation of motion for material point x_k (see Equation 3.8), Lagrangian needs to be written in an expanded form by showing only the terms associated with the material point x_k

$$L = \dots + \frac{1}{2} \rho_k \dot{\mathbf{u}}_k \dot{\mathbf{u}}_k V_k - \sum_{j=1}^{\infty} \frac{1}{2} [w_{kj}(\mathbf{y}_m - \mathbf{y}_k) + w_{jk}(\mathbf{y}_n - \mathbf{y}_j)] V_k - (\mathbf{b}_k \mathbf{u}_k) V_k + \dots \quad (3.14)$$

Using this form of Lagrangian, first term on LHS of Equation 3.8 can be easily determined

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{u}}_k} \right) = \rho_k \ddot{\mathbf{u}}_k V_k \quad (3.15)$$

In contrast to the first term on LHS of Equation 3.8 second term is slightly harder to solve as it requires careful application of the chain rule. If more in-depth explanation is needed, reader can refer to Madenci and Oterkus (2014) book.

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{u}_k} = & \left[\sum_{j=1}^{\infty} \frac{1}{2} \left(\sum_{m=1}^{\infty} \frac{\partial w_{km}}{\partial (\mathbf{y}_j - \mathbf{y}_k)} V_m \right) \frac{\partial (\mathbf{y}_j - \mathbf{y}_k)}{\partial \mathbf{u}_k} + \right. \\ & \left. \sum_{j=1}^{\infty} \frac{1}{2} \left(\sum_{n=1}^{\infty} \frac{\partial w_{nk}}{\partial (\mathbf{y}_k - \mathbf{y}_j)} V_n \right) \frac{\partial (\mathbf{y}_k - \mathbf{y}_j)}{\partial \mathbf{u}_k} - \mathbf{b}_k \right] V_k \end{aligned} \quad (3.16)$$

As it can be seen from the Equation 3.16 there are additional summation terms $\sum_{m=1}^{\infty}$ and $\sum_{n=1}^{\infty}$. These terms come from the fact that w_{km} and w_{nk} are functions of all bonds/interactions in their respective families. By substituting LHS terms in Equation 3.8 with Equation 3.15 and Equation 3.16 and rewriting it in a slightly more under-

standable form a standard equation of motion can be obtained:

$$\rho_k \ddot{\mathbf{u}}_k = \sum_{j=1}^{\infty} \frac{1}{2} \left(\sum_{m=1}^{\infty} \frac{\partial w_{km}}{\partial(\mathbf{y}_j - \mathbf{y}_k)} V_m \right) - \sum_{j=1}^{\infty} \frac{1}{2} \left(\sum_{n=1}^{\infty} \frac{\partial w_{nk}}{\partial(\mathbf{y}_k - \mathbf{y}_j)} V_n \right) + \mathbf{b}_k \quad (3.17)$$

According to Madenci and Oterkus (2014), $\sum_{m=1}^{\infty} \frac{\partial w_{km}}{\partial(\mathbf{y}_j - \mathbf{y}_k)} V_m$ represents the force density that material point x_j exerts on material point x_k and $\sum_{n=1}^{\infty} \frac{\partial w_{nk}}{\partial(\mathbf{y}_k - \mathbf{y}_j)} V_n$ represents the force density that material point x_k exerts on material point x_j . By introducing force density vectors, Equation 3.17 is transformed into a standard nonordinary form of peridynamic equation of motion, see Figure 3.3

$$\rho(\mathbf{x}_k) \ddot{\mathbf{u}}_k = \sum_{j=1}^{\infty} (\mathbf{t}_{kj}(\mathbf{u}_j - \mathbf{u}_k, \mathbf{x}_j - \mathbf{x}_k, t) - \mathbf{t}_{jk}(\mathbf{u}_k - \mathbf{u}_j, \mathbf{x}_k - \mathbf{x}_j, t)) V_j + \mathbf{b}_k \quad (3.18)$$

where force density vector \mathbf{t}_{kj} is expressed as

$$\mathbf{t}_{kj}(\mathbf{u}_j - \mathbf{u}_k, \mathbf{x}_j - \mathbf{x}_k, t) = \frac{1}{2} \frac{1}{V_j} \left(\sum_{m=1}^{\infty} \frac{\partial w_{km}}{\partial(\mathbf{y}_j - \mathbf{y}_k)} V_m \right) \quad (3.19)$$

and force density vector \mathbf{t}_{jk} as

$$\mathbf{t}_{jk}(\mathbf{u}_k - \mathbf{u}_j, \mathbf{x}_k - \mathbf{x}_j, t) = \frac{1}{2} \frac{1}{V_j} \left(\sum_{n=1}^{\infty} \frac{\partial w_{nk}}{\partial(\mathbf{y}_k - \mathbf{y}_j)} V_n \right) \quad (3.20)$$

Equations 3.19 and 3.20 can be expressed by means of strain energy densities of material points k and j , which will be of great help when going into ordinary and bond based peridynamics

$$\mathbf{t}_{kj} = \frac{1}{V_j} \frac{\partial \left(\frac{1}{2} \sum_m w_{km} V_m \right)}{\partial(\mathbf{y}_j - \mathbf{y}_k)} = \frac{1}{V_j} \frac{\partial W_k}{\partial(\mathbf{y}_j - \mathbf{y}_k)} \quad (3.21)$$

$$\mathbf{t}_{jk} = \frac{1}{V_j} \frac{\partial \left(\frac{1}{2} \sum_n w_{nk} V_n \right)}{\partial(\mathbf{y}_k - \mathbf{y}_j)} = \frac{1}{V_j} \frac{\partial W_j}{\partial(\mathbf{y}_k - \mathbf{y}_j)} \quad (3.22)$$

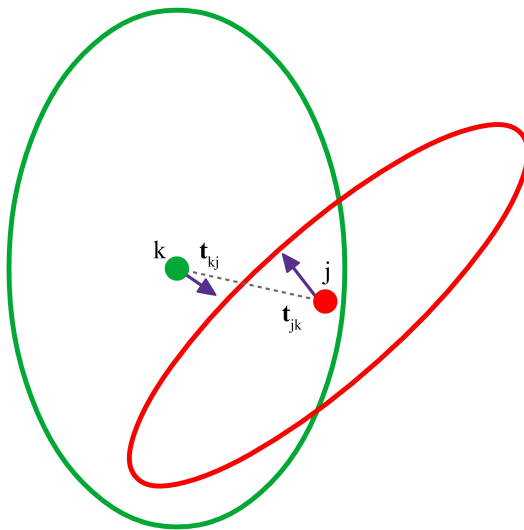


Figure 3.3: Nonordinary State Based Peridynamics

3.2.1 Ordinary State Based Peridynamics

Ordinary state based peridynamic formulation was first introduced by Silling et al. (2007) which was preceded by much simpler formulation called bond based (Silling (2000)) which will be explained in the following Chapter 3.2.2. As shown in Figure 3.4, the force density vectors have unequal magnitudes while being parallel and opposite to each other and according to Madenci and Oterkus (2014) also satisfy the requirement for balance of angular momentum. Thus, they can be defined as follows

$$\mathbf{t}_{kj} = \frac{1}{2}A \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \quad (3.23)$$

and

$$\mathbf{t}_{jk} = \frac{1}{2}B \frac{\mathbf{y}_k - \mathbf{y}_j}{|\mathbf{y}_k - \mathbf{y}_j|} = -\frac{1}{2}B \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \quad (3.24)$$

Parameters A and B in the Equation 3.23 and 3.24 are parameters dependent on material type, deformation field, and the horizon. As it was mentioned above the term ordinary state based was first introduced by Silling et al. (2007). Furthermore according to Silling et al. (2007) ordinary state based formulation of the force density vectors permits decoupled distortional and volumetric deformations and it enables the

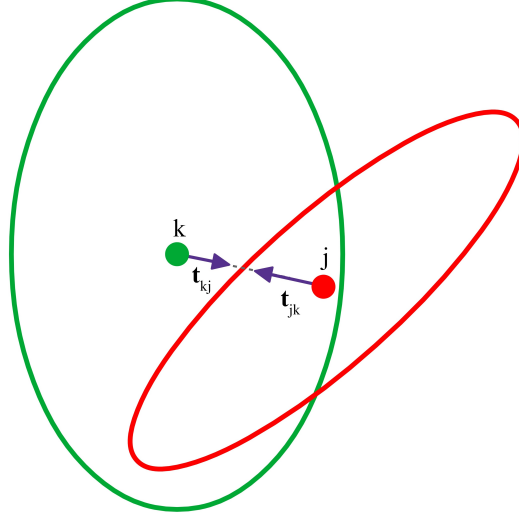


Figure 3.4: Ordinary State Based Peridynamics

enforcement of plastic incompressibility. Parameters A and B can be determined by using the relationship from Equation 3.21 and 3.22, although now these equations need to be written in a more appropriate form,

$$\mathbf{t}_{kj} = \frac{1}{V_j} \frac{\partial W_k}{\partial (|\mathbf{y}_j - \mathbf{y}_k|)} \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \quad (3.25)$$

and

$$\mathbf{t}_{jk} = \frac{1}{V_j} \frac{\partial W_j}{\partial (|\mathbf{y}_k - \mathbf{y}_j|)} \frac{\mathbf{y}_k - \mathbf{y}_j}{|\mathbf{y}_k - \mathbf{y}_j|} \quad (3.26)$$

where $\frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|}$ and $\frac{\mathbf{y}_k - \mathbf{y}_j}{|\mathbf{y}_k - \mathbf{y}_j|}$ represent unit vector or direction of force density vector. Comparing Equation 3.23 and 3.24 with Equation 3.25 and 3.26 it can be seen that parameters A and B can be determined if appropriate form of strain energy density is defined, W_k and W_j . According to Madenci and Oterkus (2014), for isotropic and elastic material, the explicit form of the strain energy density can be expressed as

$$W_k = a\Theta_k^2 - a_2\Theta_k T_k + a_3 T_k^2 + b \sum_{j=1}^N \omega_{kj} \left((|\mathbf{y}_j - \mathbf{y}_k| - |\mathbf{x}_j - \mathbf{x}_k|) - \alpha T_k |\mathbf{x}_j - \mathbf{x}_k| \right)^2 V_j \quad (3.27)$$

and

$$W_j = a\Theta_j^2 - a_2\Theta_j T_j + a_3 T_j^2 + b \sum_{j=1}^N \omega_{jk} \left((|\mathbf{y}_k - \mathbf{y}_j| - |\mathbf{x}_k - \mathbf{x}_j|) - \alpha T_j |\mathbf{x}_k - \mathbf{x}_j| \right)^2 V_j \quad (3.28)$$

In Equation 3.27 $\omega_{kj} = \omega(|\mathbf{x}_j - \mathbf{x}_k|)$ is the nondimensional influence or weight function that determines the influence of family points on the main material point k . Weight function ω is usually defined as

$$\omega_{kj} = \frac{\delta}{|\mathbf{x}_j - \mathbf{x}_k|} \quad (3.29)$$

Furthermore T_k and α represent temperature at material point k and coefficient of thermal expansion. The same can be stated for material point j in Equation 3.28. Θ_k and Θ_j are the dilatation terms and they have the following expression

$$\Theta_k = d \sum_{m=1}^N w_{km} (s_{km} - \alpha T_k) \frac{\mathbf{y}_m - \mathbf{y}_k}{|\mathbf{y}_m - \mathbf{y}_k|} (\mathbf{x}_m - \mathbf{x}_k) V_m + 3\alpha T_k \quad (3.30)$$

and

$$\Theta_j = d \sum_{n=1}^N w_{jn} (s_{jn} - \alpha T_j) \frac{\mathbf{y}_n - \mathbf{y}_j}{|\mathbf{y}_n - \mathbf{y}_j|} (\mathbf{x}_n - \mathbf{x}_k) V_n + 3\alpha T_j \quad (3.31)$$

where s_{km} is the stretch definition of a bond in peridynamic theory and is expressed as

$$s_{km} = \frac{|\mathbf{y}_m - \mathbf{y}_k| - |\mathbf{x}_m - \mathbf{x}_k|}{|\mathbf{x}_m - \mathbf{x}_k|} \quad (3.32)$$

Expression for s_{jn} is the same with only different indexes. In above equations peridynamic parameter d ensures that Θ_k and Θ_j remain nondimensional. Furthermore peridynamic material parameters, a , a_2 , a_3 , and b , in Equations 3.26 and 3.27 can be related to the material constants from classical continuum mechanics by considering simple loading conditions. In-depth explanation of the procedure can be found in Madenci and Oterkus (2014), where material parameters are expressed for 1D, 2D and

3D structures as

$$\begin{aligned}
 a &= \frac{1}{2} \left(\kappa - \frac{5\mu}{3} \right), & a_2 &= 6\alpha a, & a_3 &= 9\alpha^2 a, & b &= \frac{15\mu}{2\pi\delta^5} & d &= \frac{9}{4\pi\delta^4} & \text{for } 3D \\
 a &= \frac{1}{2} (\kappa - 2\mu), & a_2 &= 4\alpha a, & a_3 &= 4\alpha^2 a, & b &= \frac{6\mu}{\pi h\delta^4} & d &= \frac{2}{\pi h\delta^3} & \text{for } 2D \\
 a &= a_2 = a_3 = 0, & b &= \frac{E}{2A\delta^3} & d &= \frac{1}{2\delta^2 A} & \text{for } 1D
 \end{aligned} \tag{3.33}$$

where A is cross-sectional area and h is thickness. After combining Equation 3.25, 3.27 and 3.30 and performing differentiation, parameter A in the force density vector (Equation 3.23 can be rewritten in terms of peridynamic material parameters as

$$\begin{aligned}
 A &= 4w_{kj} \left[d \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \frac{\mathbf{x}_j - \mathbf{x}_k}{|\mathbf{x}_j - \mathbf{x}_k|} \left(\alpha\Theta_k - \frac{1}{2}a_2T_k \right) \right. \\
 &\quad \left. + b \left((|\mathbf{y}_j - \mathbf{y}_k| - |\mathbf{x}_j - \mathbf{x}_k|) - \alpha T_k |\mathbf{x}_j - \mathbf{x}_k| \right) \right]
 \end{aligned} \tag{3.34}$$

Exact same procedure can be applied for parameter B by combining Equation 3.26, 3.28 and 3.31 and can be expressed as

$$\begin{aligned}
 B &= 4w_{jk} \left[d \frac{\mathbf{y}_k - \mathbf{y}_j}{|\mathbf{y}_k - \mathbf{y}_j|} \frac{\mathbf{x}_k - \mathbf{x}_j}{|\mathbf{x}_k - \mathbf{x}_j|} \left(\alpha\Theta_j - \frac{1}{2}a_2T_j \right) \right. \\
 &\quad \left. + b \left((|\mathbf{y}_k - \mathbf{y}_j| - |\mathbf{x}_k - \mathbf{x}_j|) - \alpha T_j |\mathbf{x}_k - \mathbf{x}_j| \right) \right]
 \end{aligned} \tag{3.35}$$

Now force density vectors t_{kj} and t_{jk} can be rewritten as

$$\mathbf{t}_{kj} = 2\delta \left[d \frac{\Lambda_{kj}}{|\mathbf{x}_j - \mathbf{x}_k|} \left(a\Theta_k - \frac{1}{2}a_2T_k \right) + b(s_{kj} - \alpha T_k) \right] \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \tag{3.36}$$

and

$$\mathbf{t}_{jk} = 2\delta \left[d \frac{\Lambda_{jk}}{|\mathbf{x}_k - \mathbf{x}_j|} \left(a\Theta_j - \frac{1}{2}a_2T_j \right) + b(s_{jk} - \alpha T_k) \right] \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \tag{3.37}$$

where parameter Λ_{kj} , is defined as

$$\Lambda_{kj} = \left(\frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \right) \cdot \left(\frac{\mathbf{x}_j - \mathbf{x}_k}{|\mathbf{x}_j - \mathbf{x}_k|} \right) \tag{3.38}$$

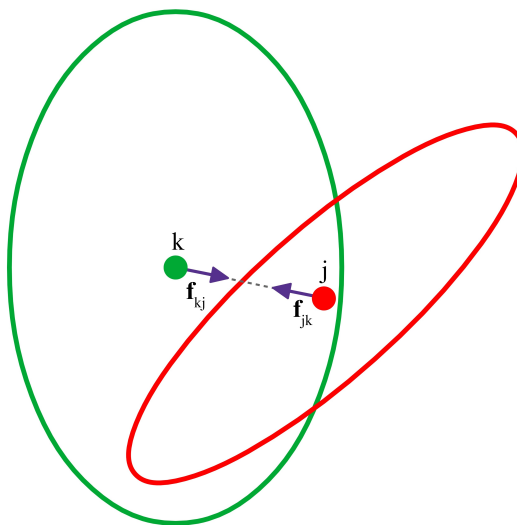


Figure 3.5: Bond Based Peridynamics

3.2.2 Bond Based Peridynamics

Bond based peridynamics is a special case formulated by Silling (2000), where force density vectors are not only parallel to the relative position vector in the deformed state as in ordinary state based peridynamics, but are also equal in magnitude (see Figure 3.5). These assumptions mean that force density vector can be expressed in the form

$$\mathbf{t}_{kj} = \frac{1}{2}C \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} = \frac{1}{2}\mathbf{f}_{kj} \quad (3.39)$$

and

$$\mathbf{t}_{jk} = \frac{1}{2}C \frac{\mathbf{y}_k - \mathbf{y}_j}{|\mathbf{y}_k - \mathbf{y}_j|} = -\frac{1}{2}C \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} = -\frac{1}{2}\mathbf{f}_{kj} \quad (3.40)$$

similarly to the ordinary state based, parameter C in Equation 3.39 and 3.40 is a parameter that depends on the material type, stretch between \mathbf{x}_k and \mathbf{x}_j , and the horizon. Also \mathbf{f}_{kj} is known as the pairwise force function.

In order to satisfy above mentioned assumptions for the force density vectors, every bond needs to be independent from influence of other bonds, meaning that micropotential of a bond only depends on deformation from the family points. This leads to

the following expression of a micropotential

$$\begin{aligned} w_{kj} &= w_{kj}(\mathbf{y}_j - \mathbf{y}_k) \\ w_{jk} &= w_{jk}(\mathbf{y}_k - \mathbf{y}_j) \end{aligned} \quad (3.41)$$

Because definition of micropotentials is different from Equation 3.5, this means that new equation of motion needs to be formulated. To do this same procedure as before needs to be followed. Lagrangian needs to be written in an expanded form by showing only the terms associated with the material point x_k

$$L = \dots + \frac{1}{2} \rho_k \dot{\mathbf{u}}_k \dot{\mathbf{u}}_k V_k - \sum_{j=1}^{\infty} \frac{1}{2} [w_{kj}(\mathbf{y}_j - \mathbf{y}_k) + w_{jk}(\mathbf{y}_k - \mathbf{y}_j)] V_k - (\mathbf{b}_k \mathbf{u}_k) V_k + \dots \quad (3.42)$$

Using this form of Lagrangian, first term on LHS of Equation 3.8 can be easily determined

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{u}}_k} \right) = \rho_k \ddot{\mathbf{u}}_k V_k \quad (3.43)$$

Second term on LHS of Equation 3.8 is now much easier to solve because bond micropotentials only depend on deformation from family points

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{u}_k} &= \left[\frac{1}{2} \sum_{j=1}^{\infty} \frac{\partial w_{kj}}{\partial (\mathbf{y}_j - \mathbf{y}_k)} \frac{\partial (\mathbf{y}_j - \mathbf{y}_k)}{\partial \mathbf{u}_k} V_j + \right. \\ &\quad \left. \frac{1}{2} \sum_{j=1}^{\infty} \frac{\partial w_{jk}}{\partial (\mathbf{y}_k - \mathbf{y}_j)} \frac{\partial (\mathbf{y}_k - \mathbf{y}_j)}{\partial \mathbf{u}_k} V_j - \mathbf{b}_k \right] V_k \end{aligned} \quad (3.44)$$

After sorting out Equation 3.44, simplified equation of motion for a material point k can be expressed as

$$\rho_k \ddot{\mathbf{u}}_k = \frac{1}{2} \sum_{j=1}^{\infty} \frac{\partial w_{kj}}{\partial (\mathbf{y}_j - \mathbf{y}_k)} V_j - \frac{1}{2} \sum_{j=1}^{\infty} \frac{\partial w_{jk}}{\partial (\mathbf{y}_k - \mathbf{y}_j)} V_j - \mathbf{b}_k \quad (3.45)$$

Comparing to the Equation 3.21 and 3.22 it's apparent that force density vectors in case of bond based peridynamics are equal and opposite to each other and are defined

as

$$\begin{aligned} \mathbf{t}_{kj} &= \frac{1}{2} \frac{\partial w_{kj}}{\partial (\mathbf{y}_j - \mathbf{y}_k)} \\ \mathbf{t}_{jk} &= \frac{1}{2} \frac{\partial w_{jk}}{\partial (\mathbf{y}_k - \mathbf{y}_j)} = -\frac{1}{2} \frac{\partial w_{jk}}{\partial (\mathbf{y}_j - \mathbf{y}_k)} \end{aligned} \quad (3.46)$$

From the Equation 3.46 it can be seen that following equality holds

$$\mathbf{t}_{kj} = -\mathbf{t}_{jk} \quad \text{and} \quad \mathbf{f}_{kj} = 2\mathbf{t}_{kj} \quad (3.47)$$

and proves the assumption made in Equation 3.39 and 3.40. Substituting Equation 3.47 into 3.18 leads to familiar equation of motion proposed by Silling (2000)

$$\rho_k \ddot{\mathbf{u}}_k = \sum_{j=1}^{\infty} \mathbf{f}_{kj} V_j + \mathbf{b}_k \quad (3.48)$$

In order to determine parameter C in Equation 3.39 and 3.40, following comparison to the parameters A and B from ordinary state based formulation can be made. It is clear that A and B need to be equal to each other in order to satisfy Equation 3.39 and 3.40. Therefore, the terms associated with Θ_k and Θ_j in Equation 3.34 and 3.35 must vanish, thus requiring that material parameters a and d to become 0. In that case parameter C will become

$$C = 4b\omega_{kj} \left[(|\mathbf{y}_j - \mathbf{y}_k| - |\mathbf{x}_j - \mathbf{x}_k|) - \alpha T_k |\mathbf{x}_j - \mathbf{x}_k| \right] \quad (3.49)$$

Now pairwise function \mathbf{f}_{kj} can be expressed as

$$\mathbf{f}_{kj} = 4b\delta(s_{kj} - \alpha T_k) \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \quad (3.50)$$

For bond based peridynamics most used parameter is the bond constant, c and it can be defined as

$$c = 4b\delta \quad (3.51)$$

Substituting Equation 3.51 into Equation 3.50 will result in pairwise force function

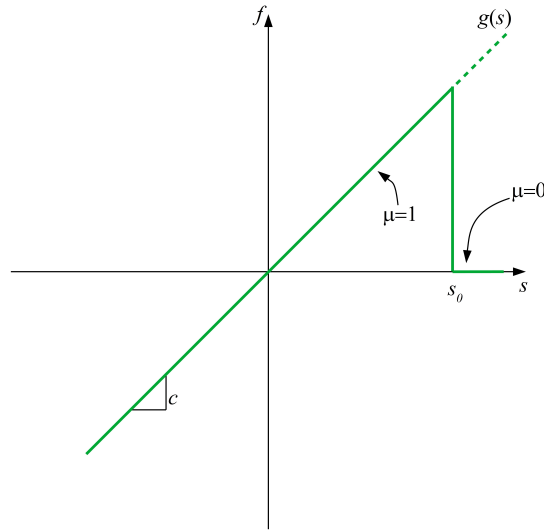


Figure 3.6: Bond force as a function of bond stretch

introduced by Silling and Askari (2005)

$$\mathbf{f}_{kj} = c(s_{kj} - \alpha T_k) \frac{\mathbf{y}_j - \mathbf{y}_k}{|\mathbf{y}_j - \mathbf{y}_k|} \quad (3.52)$$

Bond constant c can be obtained by using Equation 3.51 and 3.33

$$\begin{aligned} c &= \frac{30\mu}{\pi\delta^4} \quad \text{with } \nu = \frac{1}{4} \quad \text{for } 3D \\ c &= \frac{24\mu}{\pi h\delta^3} \quad \text{with } \nu = \frac{1}{3} \quad \text{for } 2D \\ c &= \frac{2E}{A\delta^2} \quad \text{for } 1D \end{aligned} \quad (3.53)$$

3.2.3 Introducing damage into Peridynamics

The most straightforward way to introduce failure into a peridynamic model is to allow bonds to fail after the stretch s in the bond reaches a critical limit, s_0 . Once the bond is broken, it is broken forever which means that the model is history dependent. In order to illustrate the mechanisms behind peridynamic failure, a prototype microelastic brittle material (PMB) can be used (Silling and Askari (2005)). PMB is defined as

$$f(y(t), \mathbf{x}' - \mathbf{x}) = g(s(t, \mathbf{x}' - \mathbf{x}))\mu(t, \mathbf{x}' - \mathbf{x}) \quad (3.54)$$

Linear form of force stretch relationship, is shown in Figure 3.6, where $g(s)$ is the linear scalar-valued function given by

$$g(s) = cs \quad \forall s \quad (3.55)$$

where c is a constant and μ is a history-dependent scalar-valued function that can have only two values, 1 or 0

$$\mu(t, \mathbf{x}' - \mathbf{x}) = \begin{cases} 0 & \text{if } s(t', \mathbf{x}' - \mathbf{x}) > s_0 \text{ for all } 0 \leq t' \leq t \\ 1 & \text{otherwise} \end{cases} \quad (3.56)$$

In case of bond based theory, history dependent failure parameter is introduced into the pairwise force function as

$$\mathbf{f} = \mu cs \frac{\mathbf{y}' - \mathbf{y}}{|\mathbf{y}' - \mathbf{y}|} \quad (3.57)$$

Similar approach is done for ordinary state based theory, where history dependent failure parameter is introduced into the force density vector as

$$\mathbf{t}_{kj} = 2\delta \left[ad \frac{\Lambda}{|\mathbf{x}^j - \mathbf{x}_k|} \Theta_k + b\mu(\mathbf{x}^j - \mathbf{x}_k) s_{kj} \right] \frac{\mathbf{y}^j - \mathbf{y}_k}{|\mathbf{y}^j - \mathbf{y}_k|} \quad (3.58)$$

For a more detailed explanation reader can refer to the book by Madenci and Oterkus (2014). After defining failure for each bond, next step is to define local damage for a material point \mathbf{x} which will in return indicate crack formation in the body. Local damage is simply the weighted ratio of the number of broken interactions to the initial number of interactions and can be expressed as

$$\phi(\mathbf{x}, t) = 1 - \frac{\int_H \mu(\mathbf{x}' - \mathbf{x}, t) dV'}{\int_H dV'} \quad (3.59)$$

As it can be inferred from the Equation 3.59 local damage can range from 0 to 1. If local damage is 1 it means that all bonds are broken and if it is 0 it means that none of the bonds are broken.

Until now critical stretch s_0 wasn't defined. The way in which critical stretch is calculated is to equate critical energy release rate G_0 and total work required to

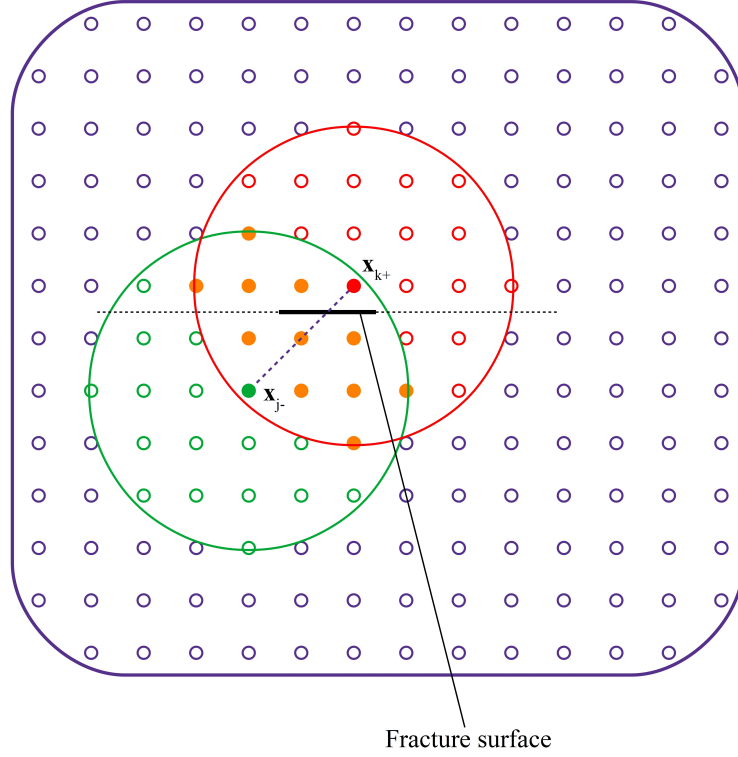


Figure 3.7: Interaction between points which cross the fracture surface

eliminate all interactions across the newly created fracture surface, see Figure 3.7. Total work or total strain energy density for ordinary state based theory can be expressed as

$$W_0 = s_c^2 \sum_{k=1}^{K^+} \sum_{j=1}^{J^-} \left[2\delta b |\mathbf{x}_{j^-} - \mathbf{x}_{k^+}| + ad^2 \delta^2 \left(\sum_{i=1}^{K^-} V_i + \sum_{i=1}^{J^+} V_i \right) \right] V_{k^+} V_{j^-} \quad (3.60)$$

Similar expression can be established for bond based peridynamics, where now $b = \frac{c}{4\delta}$ and $ad = 0$

$$W_0 = s_c^2 \sum_{k=1}^{K^+} \sum_{j=1}^{J^-} \left(\frac{1}{2} c |\mathbf{x}_{j^-} - \mathbf{x}_{k^+}| \right) V_{k^+} V_{j^-} \quad (3.61)$$

Critical energy release rate G_0 required to break all the bonds is then found from

$$G_0 = \frac{s_c^2 \sum_{k=1}^{K^+} \sum_{j=1}^{J^-} \left(\frac{1}{2} c |\mathbf{x}_{j^-} - \mathbf{x}_{k^+}| \right) V_{k^+} V_{j^-}}{A} \quad (3.62)$$

where A is the fracture area. Integral form of critical energy release rate for bond

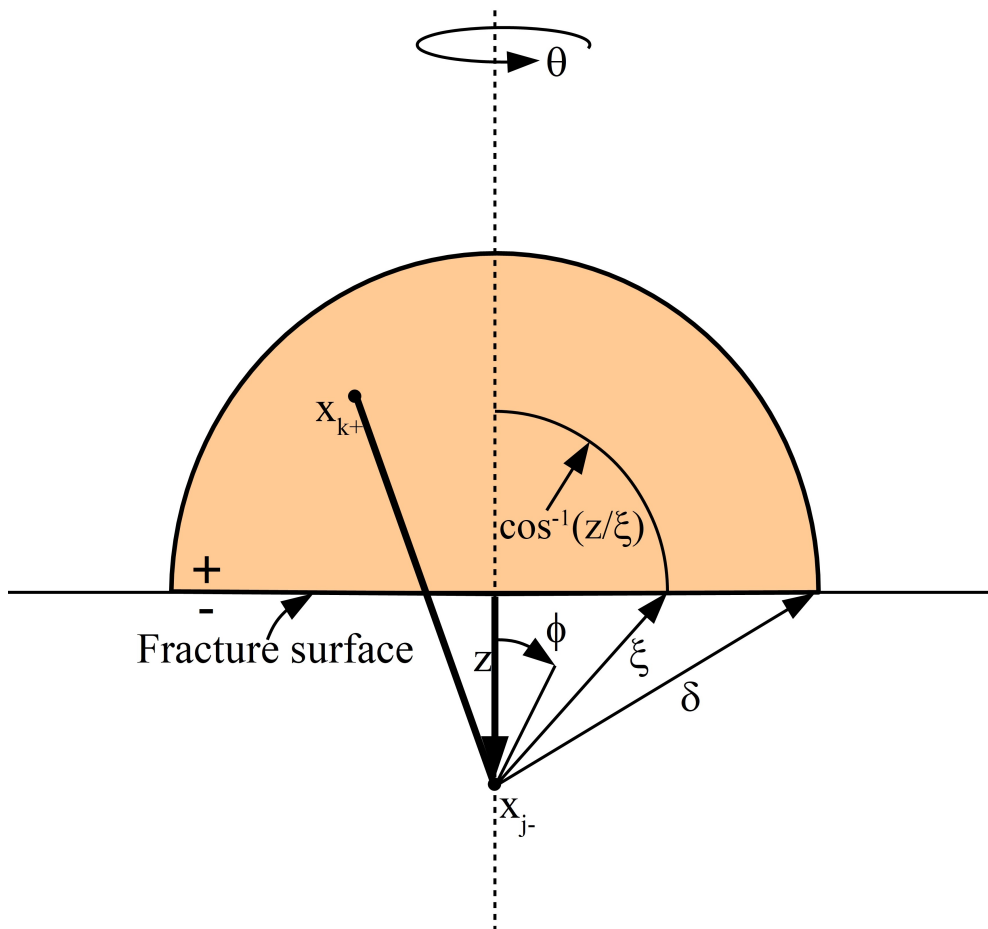


Figure 3.8: Integration domain of the micropotentials crossing a fracture surface

based peridynamics in three-dimensions was derived by Silling and Askari (2005) as

$$G_0 = \int_0^\delta \left(\int_0^{2\pi} \int_z^\delta \int_0^{\cos^{-1}\left(\frac{z}{\xi}\right)} \left(\frac{1}{2} c \xi s_0^2 \xi^2 \right) \sin \phi d\phi d\xi d\theta \right) dz \quad (3.63)$$

In essence this integral represents summation of the work required to break all of the bonds between points x_{k+} and x_{j-} , as shown in Figure 3.8. For in depth explanation reader can refer to Madenci and Oterkus (2014). After evaluating Equation 3.63, critical energy release rate for total separation of the two halves of the body can be expressed

as

$$G_0 = \begin{cases} \frac{s_0^2 c \pi \delta^5}{10} & \text{for 3D} \\ \frac{s_0^2 c h \delta^4}{4} & \text{for 2D} \end{cases} \quad (3.64)$$

Now critical stretch can be easily expressed from Equation 3.64 as

$$s_0 = \begin{cases} \sqrt{\frac{10G_0}{c\pi\delta^5}} & \text{for 3D} \\ \sqrt{\frac{4G_0}{ch\delta^4}} & \text{for 2D} \end{cases} \quad (3.65)$$

Chapter 4

Peridynamic Solver - VOPDSolver

4.1 Introduction

One of the reasons behind development of VOPDSolver (Vazic Oterkus Peridynamic Solver) was a need for a suitable programming framework in order to develop and test peridynamic codes. Up to now there are only a few peridynamic solvers (such as Peridigm or LAMMPS) that are capable of defining and solving complex tasks. Usually what happens when working on new peridynamic code is development of basic peridynamic framework. This basic framework includes simple 2D or 3D geometry and mesh development, such as a square or rectangular plate, round disc, cube or sphere. Where all of these geometrical objects are easily represented and mesh is defined by using basic trigonometry. Also simple brute force family search is usually used, which isn't suitable for large problems. Furthermore geometrically simple boundary conditions are applied and all of the supporting data is hard-coded. All of the above mentioned steps create on-off codes that aren't reusable when something is changed or at very least a lot of reprogramming is needed for code to work properly.

With this in mind VOPDSolver was created using object oriented architecture and C++ language. All the development was done in Microsoft Visual Studio IDE, target-

ing Microsoft Windows operating system. VOPDSolver has ability to define complex geometrical structures and also because it's object oriented it allows straightforward implementation of new geometrical classes. Each 2D and 3D object can be used to create peridynamic mesh (cloud of points/particles bounded by the PD body). As compared to basic peridynamic codes, VOPDSolver implements several family search algorithms: Brute-force search, Region partitioning algorithm, K-d Tree and Boost R Tree. Furthermore VOPDSolver has are several solvers which are dependant on the peridynamic theory - Bond Based Solver or Ordinary State Based Solver. Also further solver distinction is made based on problem definition - Dynamic Solver or Static/Quasi-static Solvers.

All of the above mentioned parts of VOPDSolver together with class definitions for different types of materials, boundary/loading conditions and correction factors and time loop parallelization makes VOPDSolver capable of dealing with complex static, dynamic or even impact problems. In the following sections more detailed explanation will be given for geometry and mesh classes, family search algorithms and solver types.

4.2 Geometry and Mesh

Before delving deeper into Geometry Class and Mesh Class definitions, peridynamic body and mesh definition in context of VOPDSovler needs a closer examination. Peridynamic body is a geometrical object discretised by number of particles, each describing some amount of volume. On the other hand peridynamic mesh is a cloud of points/particles bounded by the PD body. This shouldn't be conflated with finite element mesh or finite element mesh generation. In finite element case, mesh generation is a procedure where mesh algorithm generates a polygonal or polyhedral mesh that approximates a geometric domain.

Currently there are only a few peridynamic solvers (such as Peridigm or LAMMPS) that are capable of defining and solving complex tasks. One of the problems is geometry and mesh definition which becomes cumbersome as models move away from simple 2D & 3D problems (such as plate or cube). In order define, discreticize and solve complex

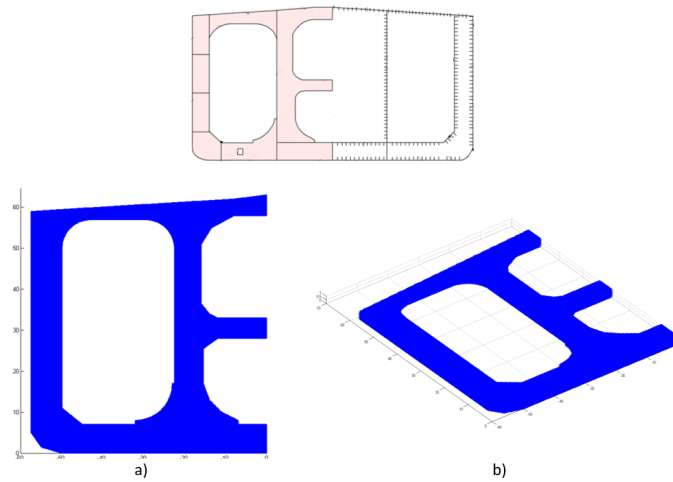


Figure 4.1: Rough peridynamic mesh of a ship hull cross section; a) 2D and b) 3D

peridynamic bodies, such as ship hull cross-section, airplane fuselage etc. (see Figure 4.1), VOPDSolver has variety of geometry classes that are capable of dealing with such problems. Because of the peculiarities of peridynamic mesh which was mentioned before, VOPDSolver geometrical objects are acting as bounding boxes. This can be easily illustrated on a 2D example of a square plate (see Figure 4.2). Mesh generation takes a few simple steps. First step is to generate an initial cloud of points that is larger than the object being discretized (in this case a plate Figure 4.2a). Second step is to superimpose the plate on top of the mesh of points (Figure 4.2b) and last step is to determine points that are inside the plate (Figure 4.2c). Once that is done the plate is fully discretized. Points that are not inside the plate are saved for later use as they could become a part of boundary condition points.

4.2.1 Geometry Definition

VOPDGeomtry classes contain definitions for variety of 2D & 3D geometric shapes, such as polygon, polyhedron, sphere, ellipse etc. Furthermore each geometry class contains mesh and material properties, as this is a common way to organize structural analysis code.

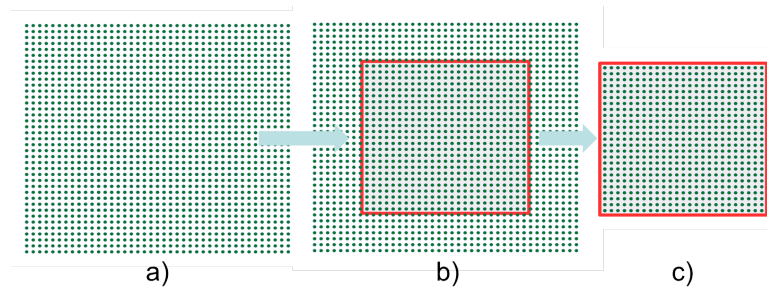


Figure 4.2: 2D peridynamic mesh generation

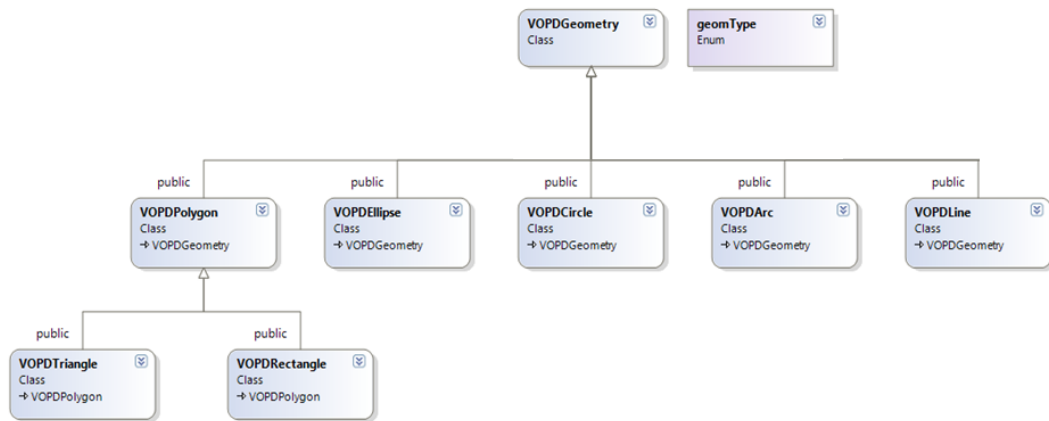


Figure 4.3: VOPDGeometry 2D class diagram

It can be seen from the Figure 4.3 and 4.4 that each geometry class inherits an abstract class called VOPDGeometry. This class contains all of the shared methods and properties that each derived geometry class needs to inherit. Those properties and methods are, for example, ability to add material, mesh, load, cut and paste boolean operations, definition of crack etc. Each derived class will implement these methods according to their own needs.

In order to generate peridynamic mesh as it is shown in Figure 4.2, each geometry class needs to implement a method that will determine the position of the point with respect to the object boundaries - is the point inside or outside of the body. For 1D objects such as lines and arcs, a simple line discretization is used as there is no need to determine if the point is inside or outside of the body. For 2D bodies a ray casting algorithm is used and for 3D bodies a simple point to plane distance algorithm is

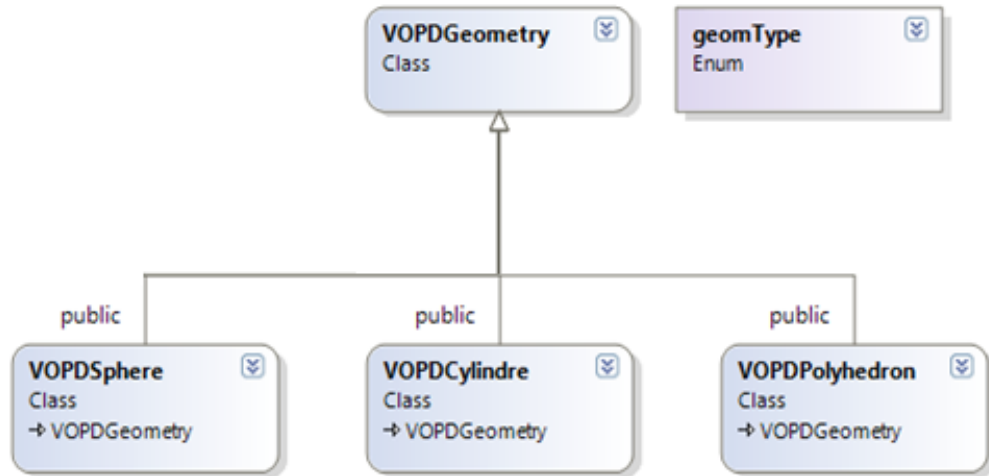


Figure 4.4: VOPDGeometry 3D class diagram

applied. Specifics of each algorithm will be explained in the following sections.

4.2.2 Mesh Definition

As it was mentioned before, in peridynamic theory a body is discretized by a finite number of material points/particles and each particle connects to other particles within a range called a material's horizon (Freimanis and Paeglitis (2017)). Those material points or the peridynamic mesh in VODPSolver is stored in a tree like structure, called spatial trees, except for the 1D mesh. For 1D peridynamic mesh, material points will be stored into a simple array in either ascending or descending order. Reason behind not using spatial trees for one dimensional spatial data is that those algorithms are developed for higher dimensional objects. Currently in VOPDSolver, K-d Tree and R-Tree are the spatial tree algorithms that are used for storing and querying peridynamic points. Main reason behind using spatial tree structures in VOPDSolver is necessity for fast and efficient family search. Furthermore spatial tree structures can be used for any other queries that can arise during peridynamic analysis, such as calculating short-range repulsive forces. Developed and used family search algorithms such as K-d

Tree and R-Tree are further explained in the following sections.

Currently in VOPDSolver, 1D, 2D, and 3D mesh classes are developed and implemented. As it can be seen from the Figure 4.5 each mesh class inherits an abstract class called VOPDMesh. This class contains all of the shared methods and properties that each derived mesh class needs to inherit. Those properties and methods are, for example, ability to add constraints, initial conditions, and different types of loads. Each derived class will implement these methods according to their own needs.

4.2.2.1 2D PD Mesh - Ray Casting Method

Ray casting algorithm for solving Point-in polygon (PIP) problem is known and used since early-1960s. In computational geometry PIP is a problem where for a given point in the plane the algorithm needs to find out if the point is inside, outside or on the boundary of a polygon. Basic idea behind Ray casting algorithm is to test how many times a ray, starting from an arbitrary point and moving in any fixed direction, intersects the edges of the polygon. There are three possible outcomes (see Figure 4.6)

- point is outside of the polygon - the ray will intersect its edge an even number of times
- point is inside of the polygon - the ray will intersect the edge an odd number of times
- point is on the edge of the polygon - in this case algorithm will fail

When the algorithm is implemented on a computer which has a finite precision arithmetic, the results may be incorrect due to rounding errors - if the point lies very close to the boundary or on the boundary (depending on the precision). Usually this isn't a concern, as speed is much more important than complete accuracy in most applications. However, in computational mechanics solver such as in Peridynamics this can potentially become a large problem. Therefore in order to create formally correct Peridynamic program, one would have to introduce a numerical tolerance ϵ and test whether PD material point lies within ϵ of the polygon edge, in which case the algorithm

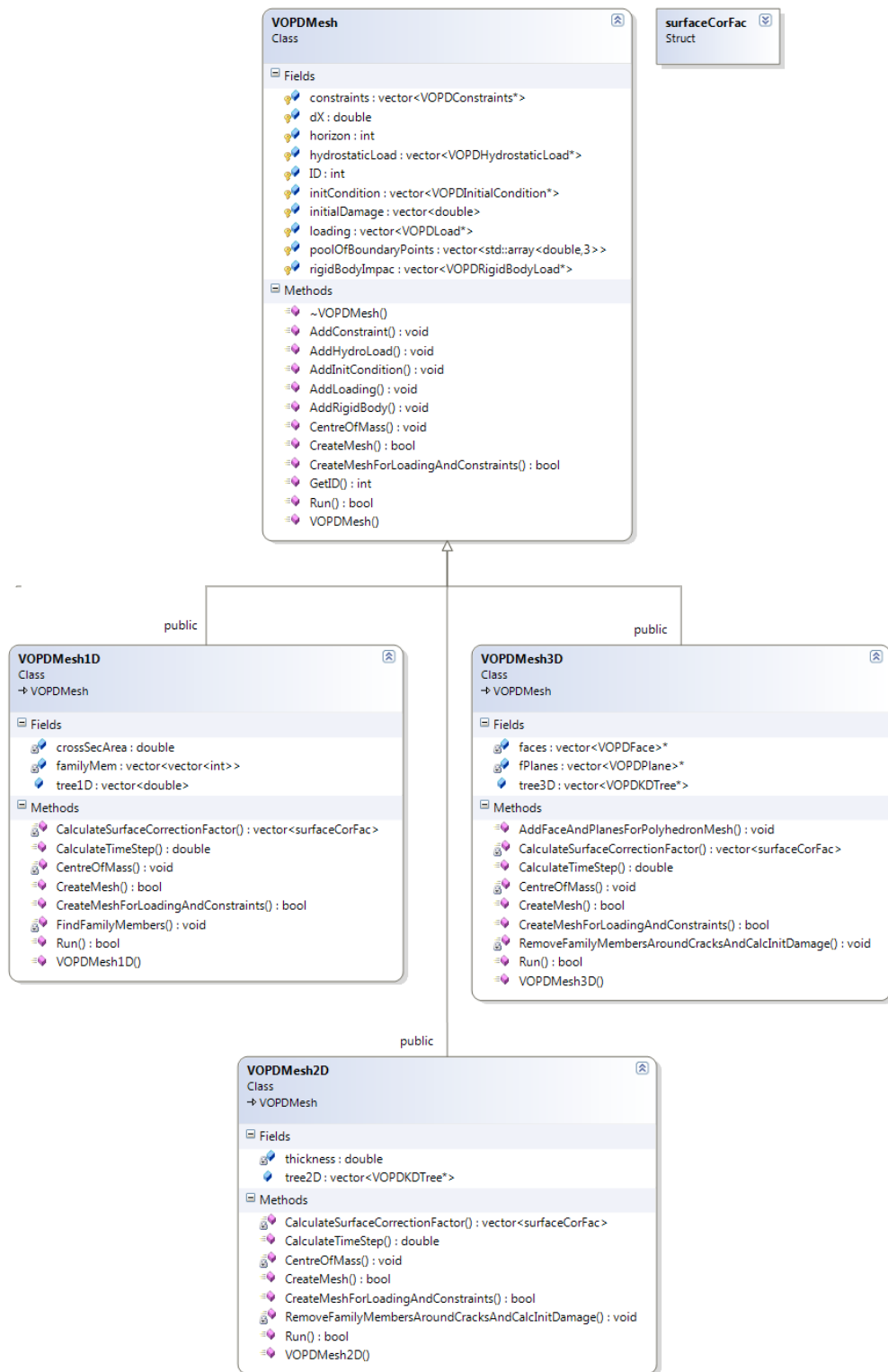


Figure 4.5: VOPDMesh class diagram

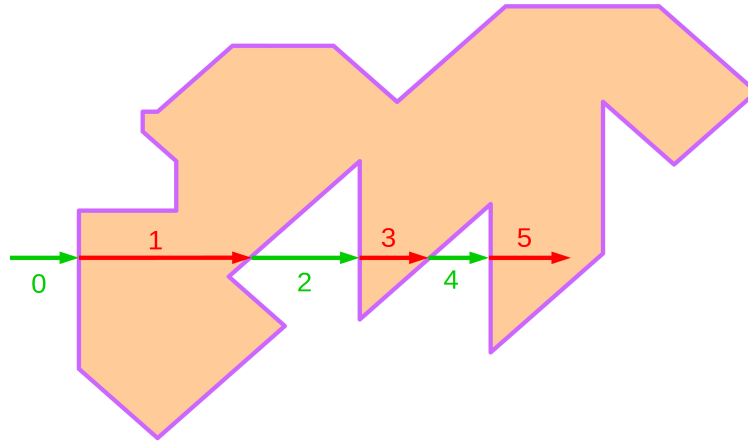


Figure 4.6: Basic idea behind Ray Casting Algorithm

should consistently include or exclude peridynamic material point from the mesh. This approach will introduce a certain numerical error that can be mitigated by increasing peridynamic mesh density.

4.2.2.2 3D PD Mesh - Point To Plane Distance Algorithm

Compared to the 2D Ray casting algorithm 3D Ray casting or Ray tracing algorithm is several times more complex and represents a monumental effort for implementation. One could be tempted to use a slightly less complex way, so called Cartesian coordinate transform, but this approach is usually used for certain special cases, such as, if the boundary is a perfect cube. This algorithm is also called Axis Aligned Cube where one can simply check if X , Y , Z coordinates of the point in consideration lie in the minimum and maximum of the X , Y , Z coordinates of the cube. There are many commercial libraries that use Ray tracing such as Blender 3D, but this was not an option as licensing was a limitation, and more simple approach was needed. To this end, a simple point to plane distance algorithm was used to determine if a the material point is inside or outside of the object. This kind of algorithm has a certain usage limitation, on account that can be only used for a convex polyhedron with N faces (see Figure 4.7). If the body is a concave object than this algorithm will not be able to determine position of the point.

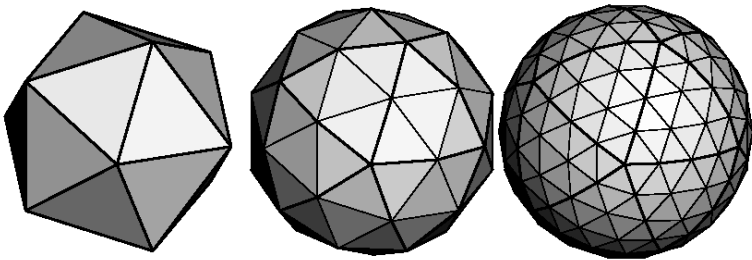


Figure 4.7: Polyhedron

Logic behind point to plane algorithm can be defined by following steps. First step is to define a polyhedron; polyhedron is a 3D body that has many faces (see Figure 4.7) and each face has a face plane in which the face lies in. For each face plane an outward normal vector is defined, which points outside of the polygon. A point to face plane distance defines a geometry vector, if the distance vector has an opposite direction with the outward normal vector, then the point is in "inside half space" of the face plane, otherwise, it is in "outside half space" of the face plane. A peridynamic point is determined to be inside of the polyhedron if the point is in "inside half space" for all faces of the polyhedron (see Figure 4.8).

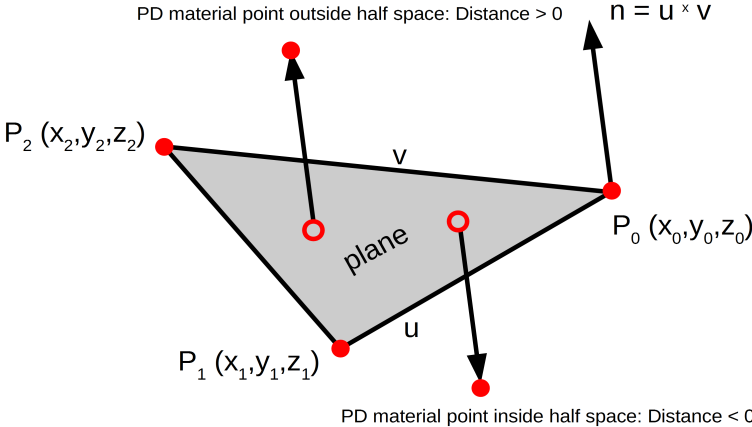


Figure 4.8: Determining if the PD point is inside or outside the polyhedron

As it was mentioned above, the distance vector is compared to the outward normal vector that is pointing outside of the polyhedron. Normal vector to the face plane is defined by the vector product of two vectors that lie in the plane. Next step is to

define that normal vector in such a way so it is pointing outwards from the polyhedron. In order to define outward normal vector for all the faces of the polyhedron following algorithm is defined as

- For any 3 vertices from N polyhedron vertices construct a triangle plane
- Generate normal vector n as a vector production of 2 edge vectors u and v from the 3 vertices P_0 , P_1 and P_2 that define the triangle. The vertice $P_0(x_0, y_0, z_0)$ is the common beginning point of the 2 edge vectors
- Calculate the face plane with equation

$$A * x + B * y + C * z + D = 0 \quad \text{where} \quad D = -(A * x_0 + B * y_0 + C * z_0) \quad (4.1)$$

- Next step is to determine if the triangle plane is a face plane by checking if all other vertice points are in the same half space of the triangle plane. The downside is that this condition requires the convex assumption, which reduces usability of this method for only convex polyhedrons. For concave polyhedrons, using this relatively simple method will make it impossible to distinguish between a polyhedron's real face and a polyhedron's inner triangle
- Find all other vertices that are in the same face plane and append them to the initial vertices
- Running this schema on all other remaining vertices the algorithm will find all faces with their outward normal vectors and their complete vertices.

One of the apparent problems with this algorithm is that this approach isnt extremely fast because in the worst case scenario - PD point is inside the polyhedron, time complexity will be $O(Nf)$, where Nf is number of faces. Never the less this algorithm is sufficient for generating relatively highly complex 3D polyhedron objects.

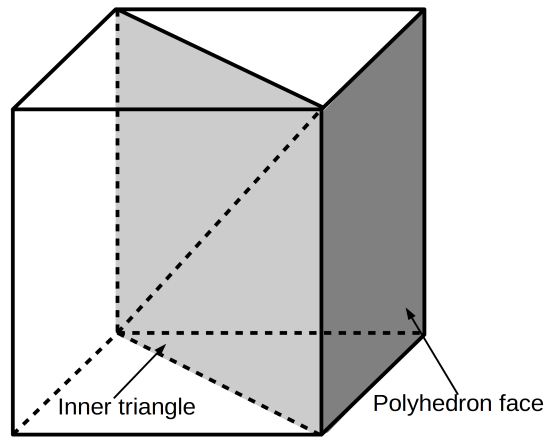


Figure 4.9: Inner triangle vs polyhedron face

4.2.3 Voronoi Tesselation for Peridynamic Mesh

When dealing with structural analysis on a micro level, material structure becomes important. This means that in peridynamic analysis polycrystalline structure of the material in question needs to be taken into a consideration. In this case Voronoi tessellation is used to represent crystallographic structure.

The Voronoi tessellation is named after Ukrainian mathematician Georgy Voronoy. Voronoi tessellation can be also found under different names, such as, Voronoi diagram, Voronoi decomposition, Voronoi partition or Dirichlet tessellation. In general Voronoi diagram is a partition of a plane into regions close to each of a given set of objects. In the simplest case, these objects are a finite set of points in the plane where for each point corresponding region consists of all the locations closer to it than to any of the other points (see Figure 4.10). These regions are called Voronoi cells, where each cell can only be a convex polygon. This means that the boundaries are defined by straight line segments and all corners have internal angles less than 180° .

Distance metric for Voronoi cell definition in engineering application is usually Euclidean distance. But that doesn't have to be the case for other disciplines, since some will use for example Manhattan distance when determining Voronoi cells. In case of VOPDSolver, Euclidean distance is the preferred option. Furthermore, when defining Voronoi cells in order to represent for example polycrystalline structure, standard

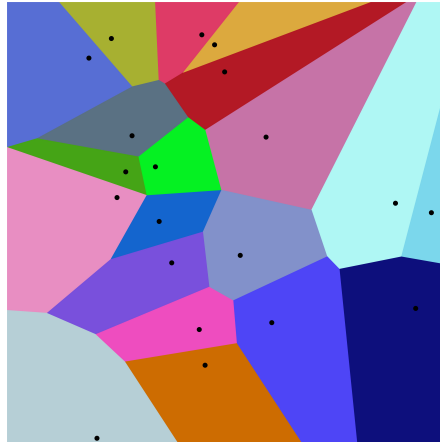


Figure 4.10: 2D Voronoi tessellation

Voronoi diagram procedure isn't necessary. This is because peridynamic mesh is composed of material points, which means that only step needed to be done is definition of random seeds (points representing crystals in a polycrystalline structure) and then finding closest material points for each seed point via Euclidean distance, see Figure 4.11. In essence this means that standard Voronoi cell is defined by a seed, edges and vertices and in case of peridynamic Voron cell it is defined only by the seed and closest material points to that seed which is the end result that is needed in order to simulate polycrystalline structure. In other words we would have the same result if we created standard Voronoi cells and than tried to find which material points belong to which Voronoi cell. This way the procedure is less computationally expensive and the accuracy is the same.

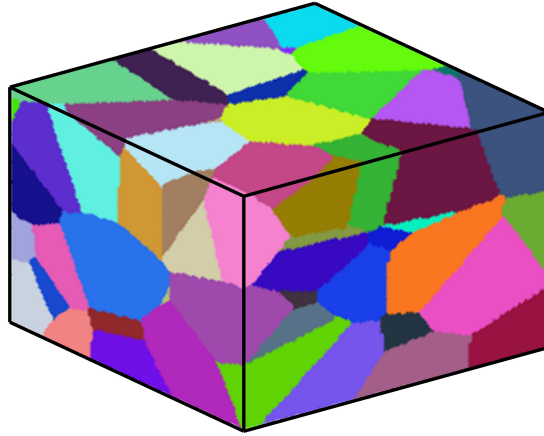


Figure 4.11: 3D Peridynamic mesh with Voronoi tessellation

4.3 Family Search

When it comes to efficiency of PD codes, apart from the time integration step, the most time consuming part is the family member search algorithm. The time-consumption is very dependent on the horizon size, δ of a material point. If PD domain is going to be solved statically, the stiffness matrix must be constructed by considering family members of each material point. Stiffness matrix created in this manner will have higher density when compared to finite element (FE) implementation of CCM. Constructing such a populated global stiffness matrix can be very time-consuming and this process, if not done in a most possible efficient way, can seriously impede the in-house PD code's efficiency. Furthermore, commonly used mesh-free methods in peridynamics experience serious issues with accuracy and convergence due to rough approximation of the contribution of family nodes close to the horizon boundary (Selsson and Littlewood (2016)). This means that when creating efficient and accurate PD codes, one needs to take into account not only family search or surface corrections, but also accurate computation of volumes near the boundary of the horizon.

The family search is basically a ranged query process where the goal is to find the members which reside inside the horizon of each material point. Although it may not be always necessary, family members of a material point need to be updated. This usually happens if adaptive search is needed.

Another factor that can influence family search is the way in which surface correction factors are calculated. As explained in Le and Bobaru (2018), there are several surface effect correction methods such as volume method, force density method, energy method, force normalization method, and fictitious nodes method. Most of these methods don't influence the family search process as they don't add any additional spatial data into the peridynamic model and mainly modify either the bond stiffness or the force state for bonds near the boundaries of the surface. Only exception is the fictitious nodes method where a layer of extra nodes (fictitious nodes) are added around surface boundaries so that every real node has a full horizon. In order for this approach to work, the size of the layer of fictitious nodes needs to be at least equal to one horizon size around the original PD domain. This extra layer of points will naturally increase the number of PD points that need to be searched and increase the family search time. According to Le and Bobaru (2018), although this approach practically eliminates PD surface effects, it has certain limitations especially when dealing with non-straight boundaries since family members definition becomes rapidly complex. On the other hand, peridynamic models with irregular and non-uniform discretized solution domains can also influence family search process (Chen (2019)). Irregular mesh can have a large impact on gridding algorithms (Verlet list, cell-linked lists or Partitioning algorithm) as it will increase unnecessary distance computations between points.

In literature, there are few studies on family member search algorithms related with Peridynamics. Diyaroglu (2016) introduced an efficient way of searching family members of each material point by utilizing localized squares for 2-dimensional (2D) and cubes for 3-dimensional (3D) configurations. Liu et al. (2018) also showed a similar family search algorithm named Family-member search with link list which utilizes an equidistant grid of squares holding certain number of points.

On the other hand, there is an extensive body of work on near-neighbour search algorithms in molecular dynamics. Methods that are predominantly investigated are Verlet List and cell-linked list. Domínguez et al. (2011) investigated the efficiency of these methods and proposed a novel neighbour search algorithm based on a dynamic updating of the Verlet List. In a study by Viccione et al. (2008), numerical sensitiv-

ity analysis of Verlet List and cell-linked list efficiency was conducted. In this work, efficiency was studied as a function of Verlet List size and cell dimensions. Another interesting study was done by Howard et al. (2016) where a novel approach based on linear bounding volume hierarchies (LBVHs) for near-neighbour search was introduced. In essence, bounding volume hierarchies (BVHs) are tree structures and mainly used in collision detection and ray tracing. They are very similar to R-tree structures that are investigated in this paper. Furthermore, these authors compared the LBVHs to the state-of-the-art algorithm based on stenciled cell lists and found that LBVHs outperformed the stenciled cell lists for systems with moderate or large size disparity and dilute or semi-dilute fractions of large particles (conditions typical in colloidal systems).

The fundamental family search algorithms available in the literature from weak to robust are investigated in this section including brute-force search, region partitioning and tree data structures. In the VOPDSolver following algorithms are implemented: Brute-force search, Region partitioning algorithm, K-d Tree and Boost R Tree. From these algorithms only K-d tree is regularly used for family search.

4.3.1 Brute-force search

The most straightforward algorithm for family member search is the so called brute-force search or exhaustive search algorithm in which all possible candidates, so called material points, are systematically enumerated. Thus, all material points, which are active in the domain, are looped over and they are checked if they satisfy a certain criterion. The criterion in this case is whether the member material point is in the range of horizon size, δ , i.e. $|\mathbf{x}_{(j)} - \mathbf{x}_{(i)}| < \delta$. As shown in Figure 4.12, if the reference length between two material points $\mathbf{x}_{(i)}$ and $\mathbf{x}_{(j)}$ or the size of a bond is bigger than the specified horizon size, the material point is skipped and other points inside the domain are checked for family members of the main material point $\mathbf{x}_{(i)}$.

It is a very simple search algorithm to implement and it always determines the correct family members of a material point. Hence, all researchers without any effort can use it to solve the problems of small size, which does not consume substantial time in family search part. The computational cost is proportional to the number of

```

for i = 1, N
  for j = 1, N
    If  $|\mathbf{x}_{(j)} - \mathbf{x}_{(i)}| < \delta$ 
      count = count + 1
      fmem(i)(count) = j

```

Figure 4.12: Brute-force search algorithm for the material points, $\mathbf{x}_{(i)}$

candidate material points and it tends to grow very quickly as the size of the problem increases, which causes combinatorial explosion. Combinatorial explosion occurs in computing environment in the following sense; if a system has n Boolean variables, which gives two possible states (true and false), the system will have 2^n possible states. If the system has n variables that can have M possible states, the system will have M^n possible states. Thus, the brute-force algorithm has the worst case complexity of $O(n^n)$, where n is the number of material points and $O(\cdot)$ represents amount of time to run the algorithm or so-called time complexity.

4.3.2 Region Partitioning

The region-partitioning algorithm proposed by Diyaroglu (2016) is elaborated in this section. In this technique, the solution domain is divided into square grids as shown in Figure 4.13. Instead of searching for the entire solution domain as in brute-force search algorithm, only the main grid, which keeps the main material point \mathbf{x} and the neighboring grids are searched for its family member points, \mathbf{x}' . It is very easy to implement and the gain in speed is substantial compared to the basic brute search. However, the oddly shaped bodies would decrease the efficiency of this algorithm. Since the grid shapes can only be in square or cubic forms, the extra time would be spent

for the points outside the main region. These points, which are outside the problem domain, must be deactivated later as shown in Figure 4.13.

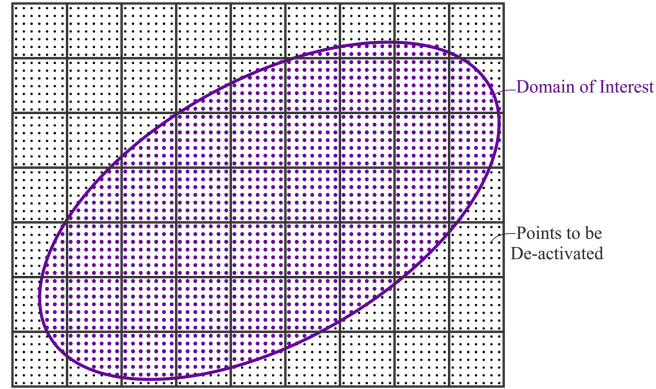


Figure 4.13: Material points inside the domain of interest with square grids

Note that Mattson and Rice (1999) proposed an approach similar to Diyaroglu (2016) to deal with near-neighbour calculations for molecular simulation techniques such as molecular dynamics or Monte Carlo. In their work, they tried to make improvement on conventional cell-linked list method as they divide the domain into a grid of cells populated by atoms and near-neighbour search was done over main cell and its neighbouring cells. In their work, they proposed a modified cell-linked list method which should substantially decrease unnecessary internuclear distance computations (neighbouring cells contain more atoms than necessary).

A rectangular problem domain with PD material points inside the square grids is shown in Figure 4.14. The size of square grids, which partition the problem domain, is determined based on the size of the horizon, δ and in this example case $\delta = 3\Delta$ in which Δ denotes the discretization size (distance between the material points). Region partitioning algorithm can be broken down into two sections including construction of material points and family member search parts. In the first section, each square region is constructed with 6 points along x and y directions except for the end regions. The region numbers are shown in red color. Thus, the family members of each main material point can only reside in its own and neighboring regions. The following scalars and arrays can be constructed accordingly;

- ncl*: Number of columns along x axis
- nrw*: Number of rows along y axis
- lstncl*: Number of points in the last column along x - axis
- lstnrw*: Number of points in the last row along y - axis
- nrngn*: Total number of regions
- region*: An array which gives the first material point's number at each region.

For a rectangular domain shown in Figure 4.14, these scalars and arrays are defined as;

$$\begin{aligned}
 ncl &= 5, nrw = 4, lstncl = 1, lstnrw = 2, nrngn = 20 \\
 region(1) &= 1, region(6) = 40, region(11) = 79, region(16) = 118 \\
 region(2) &= 10, region(7) = 49, region(12) = 88, region(17) = 124 \\
 region(3) &= 19, region(8) = 58, region(13) = 97, region(18) = 130 \\
 region(4) &= 28, region(9) = 67, region(14) = 106, region(19) = 136 \\
 region(5) &= 37, region(10) = 76, region(15) = 115, region(20) = 142
 \end{aligned}$$

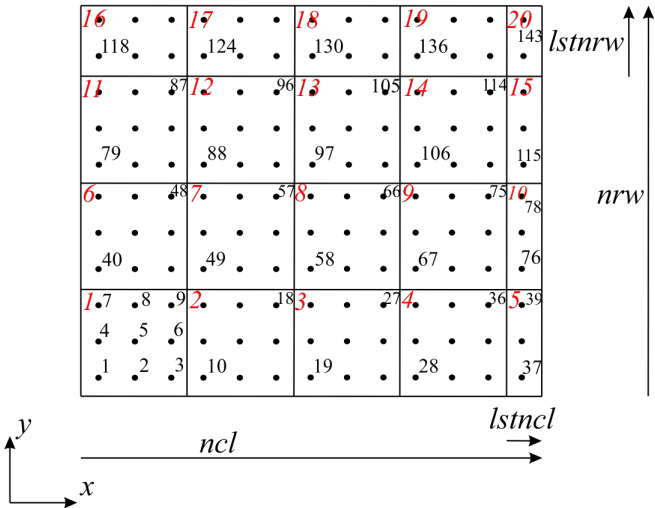


Figure 4.14: Partitioned rectangular problem domain

In the second section, the family members of each main material point are determined. Thus, the following arrays can be generated;

nfmem: Total number of family members for each main material point

fmem: Family member point numbers for each main material point

indx: Index array which defines the main material point's location in the *fmem* array

In order to create these arrays in a most efficient way and reduce the search time dramatically, the advantage of region partitioning completed in the first section is utilized. First, the main region's number and its neighboring regions numbers are defined and they are searched for family member points of the main point. For instance, if the region 14 is chosen as the main region, the search for the family members is only performed inside the neighboring regions of 8, 9, 10, 13, 15, 18, 19 and 20. Figure 4.15, shows the family member material point search for the main material point 109. The regions are enumerated locally with a blue color. At this level, the following scalars/arrays can be created;

fpoint: The first point's number (106) inside the main region.

lpoint: The last point's number (114) inside the main region.

neighrw(1:9): The number of material points along x axis at each locally numbered region;

$neighrw(1) = 3, neighrw(4) = 3, neighrw(6) = 3, neighrw(8) = 1$

$neighrw(2) = 3, neighrw(5) = 1, neighrw(7) = 3, neighrw(9) = 3$

$neighrw(3) = 1$

neighcl(1:9): The number of material points along y - axis at each locally numbered region;

$neighcl(1) = 2, neighcl(4) = 3, neighcl(6) = 3, neighcl(8) = 3$

$$\text{neighcl}(2) = 2, \text{neighcl}(5) = 3, \text{neighcl}(7) = 3, \text{neighcl}(9) = 3$$

$$\text{neighcl}(3) = 2$$

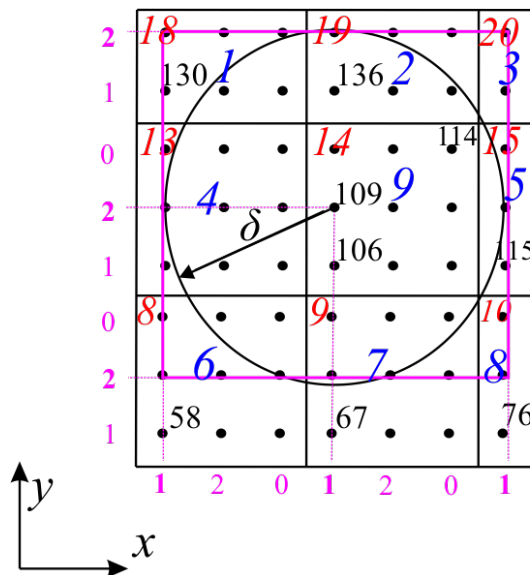


Figure 4.15: The neighboring regions for the material point 109

Thus, the family members of the main material point 109 inside the main region 14 are determined. This search algorithm can further be improved by using pink colored rectangle shown in Figure 4.15. The regions along x - and y -axes are numbered in the base of three as depicted in pink color. By doing so, the family member search is only allowed to be done in rectangular region with 7×7 points. Please see Appendix A for family search algorithm for 3-Dimensional configurations.

Although this approach gives significant amount of boost and speed in family search process, it is crude and inflexible way to organize and query the spatial data. This approach works fine for highly symmetrical meshes and straight boundaries, where the majority of the portioned regions have the optimal number of points defined by the horizon size. However, it suffers for highly irregular meshes. Furthermore, the partitioning is dependent on horizon size and any change in horizon size necessitates the repartitioning of the problem domain. These observations are self-evident when having a closer look at the algorithm, which divides the solution domain into a square grid and the family search is only done for the main region that holds the main material point

and its neighbouring grids. When using this approach, partitioning algorithm works on the assumption that it needs to search only the immediate neighbouring grids as it expects that all the family points are contained within them. This assumption comes from the fact that the mesh has an equidistant spacing between material points which isn't the case for irregular meshes. Moreover, as it was mentioned earlier, partitioning depends on the horizon size, which means that if the horizon size is changed, the partitioning needs to be updated. Updating could be done by two approaches; either number of points in each region needs to be changed or if the number of points is kept constant, then number of neighbouring regions needs to be adjusted.

4.3.3 Tree data structures

The storage and queries of peridynamic points can be achieved with many data structures. The most basic data structure is a simple array. An array is a static data structure, which can be randomly accessed and it is easy to implement as in brute-force search algorithm. On the other hand, the linked-list data structures are in essence a linear collection of data elements and each element points to the next which are dynamic in nature and are ideal for frequent operations such as adding, deleting, and updating. The main drawbacks of linked-list structures compared to static arrays are the high memory consumption and the sequentially accessed data. Other data structures including stacks, queues and hash table are specialized for complex problems.

The main disadvantage of using array or linked-list data structures to store material points is the time necessary to search for a specific point or set of points, i.e. family members. Since static arrays and linked-list structures are linear, the query time is proportional to the size of data set. This can be nicely visualized if we imagine the data set with a size of n . The number of comparisons required to find an item in the worst case scenario is $O(n)$. Therefore, efficient data structures are needed to store and search the data.

The evolved form of linked data structure (linked-list, vector, stack and queue) is a tree (Figure 4.16), which represents collection of nodes and their relations (parent-child relationship). As compared to other linear (sequential) data structures, a tree is

in non-linear or hierarchical form. A tree is either empty or comprising a root node with zero or more subtrees called children. A rooted tree form is the main interest of the current study and it has the following properties:

- One node is distinguished as the root which is node 1.
- Each node may have zero or more children.
- Every node (exterior to root) is connected with directed edge from exactly one to other node and its direction is parent to children.

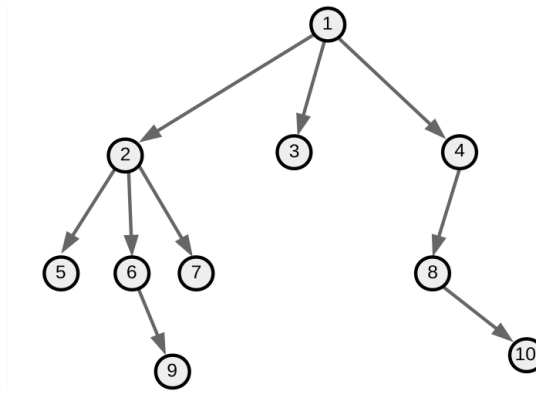


Figure 4.16: General concept of a tree structure

In Figure 4.16, node 1 is a parent (root node) and nodes 2, 3, and 4 are its children or sub-trees. On the other hand, node 2 is a parent to nodes 5, 6, and 7. Each node can have arbitrary number of children. Nodes with no children are called leaves, or external nodes. In Figure 4.16, nodes 3, 5, 7, 9, and 10 are the leaves and other nodes are called as internal nodes. Internal nodes have at least one child. Nodes with the same parent are called siblings. In Figure 4.16, nodes 2, 3, and 4 are siblings. The depth of a node is the length of the path from root to the node. For instance, the depth of node 9 is 3. The height of a node is the length of the path from node to the deepest leaf. The height of node 1 is 3. The height of a tree is equal to height of a root. The size of a node is equal to the number of nodes available in the subtree of that node (including itself). The size of node 2 is 5.

4.3.3.1 Binary Tree

Binary tree is a specialized case of general tree structure where each node has at most two children called the left and right child. If each node has exactly zero or two children, it is named as full binary tree. In a full tree, there are no nodes with exactly one child. A complete binary tree is completely filled from left to right with a possible exception of the bottom level. Figure 4.17 shows full- and complete-tree structures. A complete-tree with a height of h has between 2^h and $2^{(h+1)} - 1$.

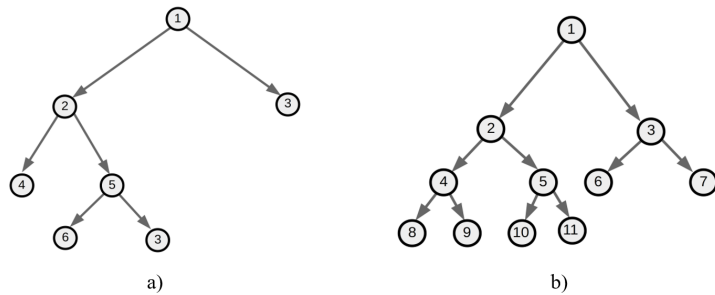


Figure 4.17: Two types of binary tree; a) full- and b) complete-tree structures

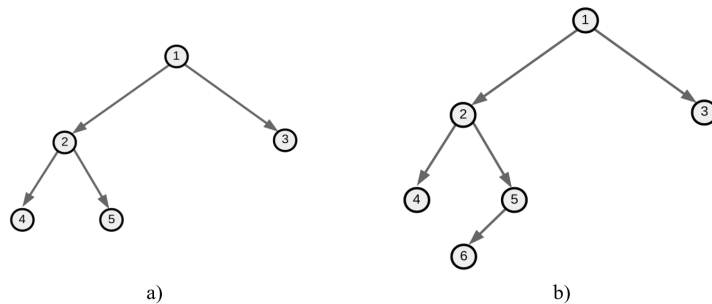


Figure 4.18: Balanced versus unbalanced binary tree; a) balanced- and b) unbalanced

Other types of binary tree are the balanced and unbalanced binary tree structures (Figure 4.18). The height of balanced-tree differs at most one from its left to the right. A balanced binary tree is also known as an AVL (Adelson Velskii Landis) tree which is developed by Adelson et al. (1962).

4.3.3.2 Binary Search Trees

A Binary Search Tree (BST) is a data structure which can be traversed/searched according to an order. A binary tree is actually a binary search tree (BST) if and only if it is in an ordered sequence. The idea of a BST is the data stored in an order so that it can be retrieved very efficiently. The nodes can be sorted as shown in (Figure 4.19) and in the following way:

- Each node contains one unique key (value used to compare nodes in case of PD this would be x,y, z position).
- The keys in the left subtree are less than the key in its parent node (L subtree).
- The keys in the right subtree are greater than the key in its parent node (R subtree).
- Duplicating node keys are not allowed.

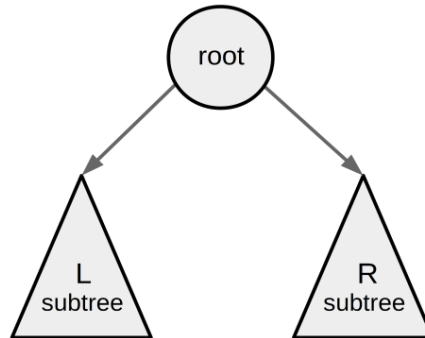


Figure 4.19: Binary Search Tree (BST) with left and right subtrees

If BST is built in a balanced form, \log time access is required for each element. In other words, algorithm needs to do at worst $\log_2(n)$ comparisons in order to find a specific node. An arbitrary BST with a height of h has total possible number of nodes equal to $2^{h+1} - 1$. In order to find a particular node only one comparison needs to be performed at each level, or a maximum of $h+1$ in total. This is because each node can only have two children and only one of them satisfies the search condition. If the number of nodes, n in a tree is known, the number of comparisons to fully traverse the

tree can be calculated as

$$2^{h+1} - 1 = n \quad (4.2)$$

which leads to

$$h = \log_2(n) - 1 = O(\log_2(n)) \quad (4.3)$$

Thus, a balanced binary search tree with n nodes has a maximum order of $\log_2(n)$ levels meaning that at most $\log_2(n)$ comparisons are needed to find a particular node. The main problem of achieving $O(\log_2(n))$ is the necessity of balanced-tree form and it is not a trivial task. One of the ways to achieve the balanced-tree form is to distribute the data randomly. The probability of forming balanced-tree structure would be high. However, if the data has already a pattern (sorted list of peridynamic points), a simple FIFO (first in first out) insertion into a binary search tree will result in growing tree either to the right or to the left side of the root node. This kind of unbalanced binary search tree is no more efficient than the regular linked-list. To this end, a great care needs to be taken in order to keep the tree as balanced as possible. There are many techniques for balancing tree structures as given in refs. Bayer (1972) and Guibas and Sedgewick (1978).

4.3.3.3 Spatial search trees

Spatial data or geospatial data is the information of a physical object which can be represented with numerical values in a geographic coordinate system. In peridynamic sense, this corresponds to material points with their volumes and positions in a coordinate system. The Geographic Information Systems (GIS) or other specialized software applications can be used to access, visualize, manipulate and analyze geospatial data.

Spatial data has two fundamental query types: nearest neighbors and range queries. Both serve as a building block for many geometric and GIS problems. Solving both problems (big data problems within a realistic time span) at a scale requires defining a spatial index. Spatial indices are used to optimize the spatial queries. Conventional

index types (binary search tree) do not efficiently handle the spatial queries; for instance, the query of the distance between two material points if they reside within the spatial area of interest. Some of the efficient spatial index methods such as R-tree and K-d tree searches can overcome this deficiency.

Data changes are usually less frequent than the queries, which means that incurring an initial time cost of processing data into an index is a fair price to pay for instant searches afterwards. This is especially true for most of the PD simulations in which initial family members do not change during the analysis.

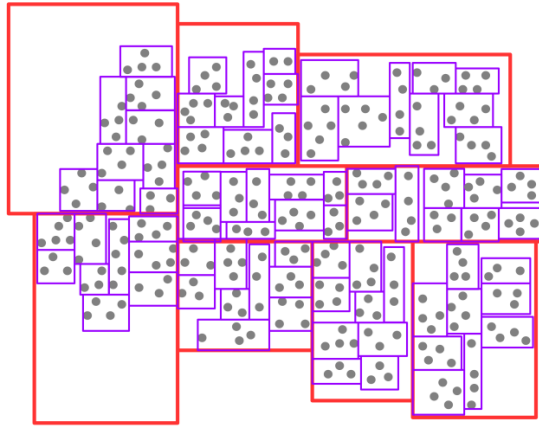


Figure 4.20: First two levels of R-tree

Almost all spatial data structures share the same principle to enable efficient search; branch and bound. This means arranging data in a tree-like structure and discarding branches if they do not fit our search criteria. The well-known spatial trees are R-tree and K-d tree. R-tree has tree data structures used for spatial access methods as proposed by Guttman (1984). The R-tree access method organizes any-dimensional data in a tree-shaped structure called an R-tree index. The index uses a bounding box which is in a rectilinear shape such that it contains the bounded objects (in case of PDs, the objects are the material points). Bounding boxes can enclose the data objects or other bounding boxes. In Figure 4.20, an R-tree with two levels of bounding boxes is shown. There are nine red boxes at the upper level and each red bounding box contains nine purple bounding boxes as the lower level. Grey points represent the peridynamic points sorted into this R-tree.

On the other hand, Bentley (1975) introduced K-d tree which is similar to R-tree. In this method, the points are sorted into two halves (around a median point) either left and right, or top and bottom, alternating between x and y, or x, y, and z or any other n-dimensions split at each level. Figure 4.21 shows the two initial splits with a red line along x-axis and subsequent splits along y-axis depicted as purple line.

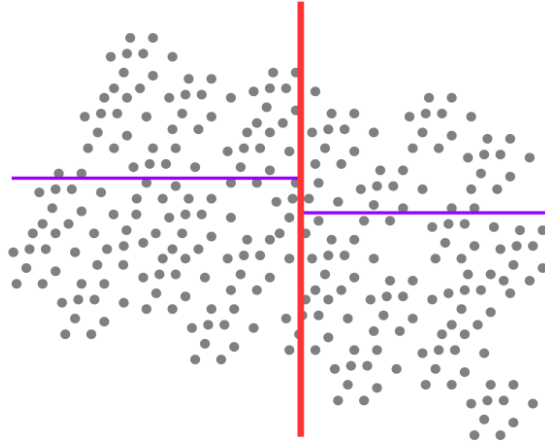


Figure 4.21: First two levels of K-d tree

Compared to R-tree, K-d tree search usually only contains points (not the rectangles) and it cannot handle the adding and removing points. However, it is easier to implement and it is usually very fast. Both R-tree and K-d tree searches share the principle of partitioning data into axis-aligned tree nodes. Since PD mesh is defined with material points which are in essence spatial data, selection of K-d or R- trees represents a logical choice when it comes to family search. In following sections, in-depth reviews of K-d tree and R-tree algorithms developed in BOOST libraries are provided and their implementation to PD codes are demonstrated.

4.3.3.4 R-tree search

R-tree is a hierarchical data structure based on B+ tree. B+ tree is a binary tree but the parent node can have more than two child nodes. R-tree is used for dynamic organization of a set of d-dimensional geometric objects (PD points can either be in 2-Dimensional or 3-Dimensional forms) and they can be represented by minimum bounding d-dimensional rectangles (MBRs). Each node of R-tree corresponds to the

MBR that bounds its children.

It must be pointed out that the MBRs surrounding different nodes may overlap with each other. Furthermore, MBR can include (in a geometrical sense) many nodes, but it can be associated with only one of them. This means that a spatial search may visit many nodes before confirming the existence of a given MBR. This also can lead to false alarms when representing geometric object with their MBRs. To avoid these kinds of mistakes, the candidate objects must be examined. For example, Figure 4.22 illustrates the case where two peridynamic material points and their horizons (red circles) which are not intersecting but their MBRs do. Therefore, R-tree represents a filtering mechanism for reduction of extremely costly direct examination of geometric objects.

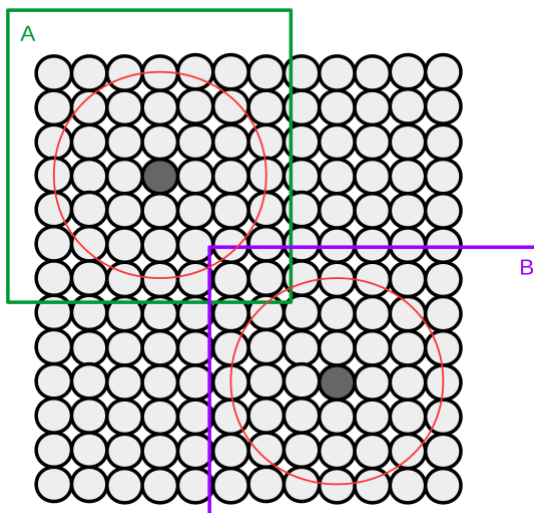


Figure 4.22: Intersecting MBRs, where peridynamic family member points are only in MBR A

An R-tree is defined by its order (n, N) and it has the following characteristics:

- Each leaf node (unless it is the root) can host up to N entries (peridynamic points), whereas the minimum allowed number of entries is $n \leq N/2$. Each entry has the form of $(\text{mbrID}, \text{oID})$, where mbrID represents the identifier of MBR that spatially contains the object and oID is the objects identifier (peridynamic point).
- Each internal node can store between $n \leq N/2$ and N entries (MBRs). Each

entry is of the form $(mbrID, p)$, where p is a pointer to the child of the node and $mbrID$ is the MBR that spatially contains the MBRs contained by child p .

- The minimum allowed number of entries in the root node is 2, unless it is a leaf. When the node is a leaf, it can contain zero or single entry because leaf nodes represent the end of a tree.
- All leaves of the R-tree are at the same level.

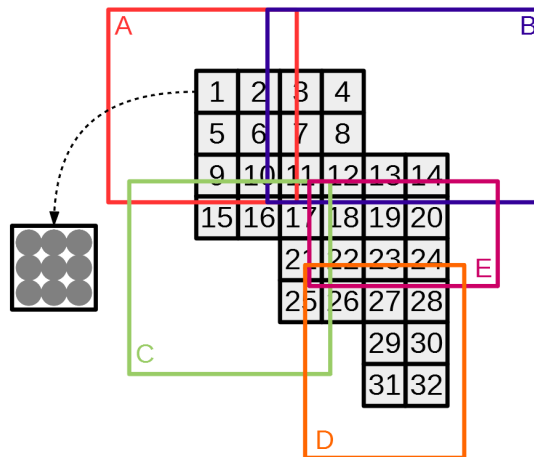


Figure 4.23: MBRs data and their inner nodes

R-tree is a height-balanced tree with all leaves are at the same level. Since, R-trees are dynamic data structures, the global re-organization does not require to handle insertions or deletions. It is one of the main advantages of R-tree compared to K-d tree and AVL tree. Figure 4.23 shows a set of MBRs with some data geometric objects. This can represent PD points. The MBRs are from number 1 to 32 and they are stored at the leaf level of R-tree. Five MBRs (A, B, C, D, and E) organize the aforementioned rectangles (where each contains 9 peridynamic points) into an internal node of R-tree. Assuming $N = 10$ and $n = 5$, Figure 4.24 depicts corresponding MBRs.

4.3.3.4.1 K-d tree search A K-d tree, or k-dimensional tree, is a binary data structure, which stores k-dimensional data, for organizing number of points in a space with k dimensions (Bentley (1975)). Each level of K-d tree splits all children with specific dimension. Each level of the tree is compared against one dimension. This means

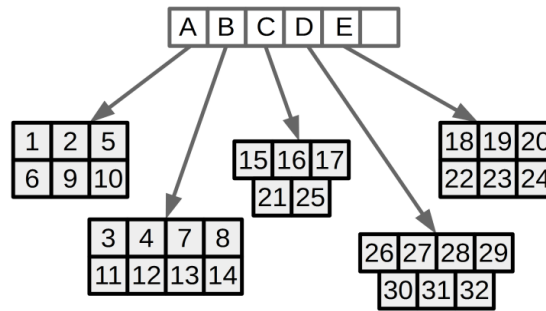


Figure 4.24: MBRs data and their inner nodes

that every node has 2 children each corresponding to an outcome of the comparison of two records based on a certain key which can be chosen as "discriminator". In a similar manner with standard binary tree, the K-d tree subdivides the data at each recursive level of the tree. Unlike standard binary tree, which uses only one key for all tree levels, the K-d tree uses k keys and it cycles through these keys for every successive tree level. In order to build 2 dimensional K-d tree (2-d tree) which comprises (x, y) coordinates, the keys would be cycled as x, y, x, y and so on for all the successive levels of K-d tree (Brown (2014)).

Figure 4.25 demonstrates the working mechanism of K-d tree. An array of points is inserted (first node in the array is a root node) to the system which eventually produces unbalanced tree. The array is given as;

$$Ar = [(8,9), (5,11), (15,10), (10,7), (5,3), (2,6), (12,4), (1,7)]$$

The first cutting plane is in the x direction (blue line) and the next cutting plane is in the y direction (red line) and so on. This process is repeated until the leaf level is reached meaning that there are no more points to insert.

4.3.3.5 Balanced K-d tree search

When building a K-d tree, due to the use of different keys at successive levels of the tree it is not possible to employ rebalancing techniques. Building K-d tree data structure would cause unbalanced structures. The reason of this is the use of different keys at successive levels of tree data. Moreover, it is not possible to employ rebalancing techniques. Rebalancing techniques are used to build self-balancing AVL tree (Adelson-

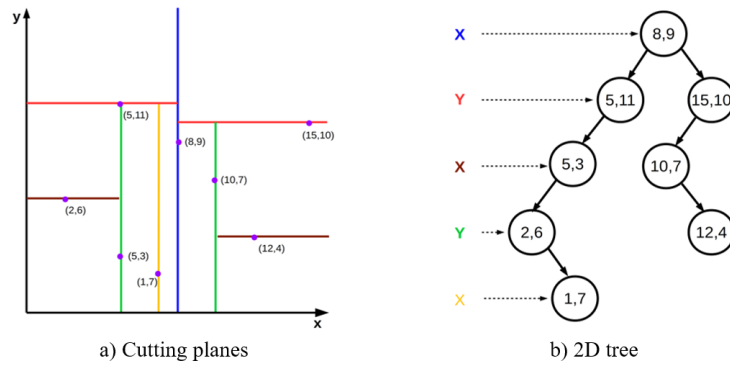


Figure 4.25: 2-d tree

Velskii and Landis), where if the height of two child subtrees of any node differ by more than one. In that case, rebalancing is performed to restore the height. Another self-balancing tree is the so called the red-black tree (Bayer (1972), Guibas and Sedgwick (1978)), where each node of the binary tree has an extra bit. This bit is often interpreted as the color (red or black) of a node. These color bits are then used to ensure that the tree remains approximately balanced during the insertions and deletions. Since it's not possible to employ rebalancing techniques, the typical approach to building a balanced K-d tree is to find the median of the data for each recursive subdivision of the data. Bentley (1975) showed that if the median of n elements is found in $O(n)$ time, it would be possible to build a depth-balanced K-d tree in $O(n \log(n))$ time. In order to find the median of n elements, sorting algorithm needs to be applied to the data. Most widely used sorting algorithms are Quicksort, Merge Sort, and Heapsort. Quicksort (Hoare (1962)) is a divide and conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. In the best case scenario, Quicksort finds the median in $O(n \log(n))$ time and in the worst case scenario the time increases up to $O(n^2)$. Merge Sort (Goldstine and von Neumann (1963)) is also a divide and conquer algorithm. The idea behind the Merge Sort is to divide the unsorted list into n sub-lists. Each sub-list contains one element and a list of one element is considered as sorted. Afterwards, Merge Sort repeatedly merges sub-lists to produce a new sorted sub-lists until there is only one sub-list remaining. On the other hand, Heap sort is a comparison based sorting technique based on Binary Heap data structure. It is similar

to selection sort which finds the maximum element. Merge sort and Heapsort find the median in the best case of $O(n \log n)$, which leads to $O(n \log^2 n)$ time for a balanced K-d tree (Wald and Havran (2006)).

An alternative approach to building a balanced K-d tree would be the presorting data prior to building a tree (Brown (2014)). The algorithms developed by Brown (2014) are implemented in our in-house PD solver. The PD points are presorted in each of k dimension prior to building K-d tree. Thus, it maintains the order of these k sorts when building a balanced K-d tree. This in return achieves a worst-case complexity of $O(kn \log n)$.

Basic concepts of balanced K-d tree algorithm can be explained with the following simple example. A small sample of spatial data is considered. This data can be viewed as a set of PD points from which a K-d Tree is created. The data set consists of 15 (x, y, z) tuples (PD points) which are stored into a list of elements numbered from 0 through 14 as shown in Figure 4.26. First step is to presort the PD points using merge sort. The points are sorted via super keys; x:y:z, y:z:x, and z:x:y which represent cyclic permutations of x, y, and z. The points are not sorted independently through x, y, and z coordinates but each part of the super key (x, y, and z) has a certain level of significance. Hence, for example, the super key y:z:x is composed from y as a primary key, z as a secondary key and x as a tertiary key. This means that during the merge sort, if the two points have identical primary keys, then they are compared using secondary key, and if their secondary keys are identical, they are compared using the tertiary key. In case of all the three keys are the same with two identical points, one of the points is removed. In order to have initial array of points untouched and to save on memory consumption, the merge sort does not work with initial array of PD points. Instead, it reorders three index arrays whose elements point to array indices. The initial order of indices produced by merge sort is shown in Figure 4.26 (see *xyz*, *yzx* and *zxy* columns under Initial Indices). The next step is to partition the points in x direction, which is the first splitting dimension, by using *x:y:z* super key. There, the partition location is specified by the median element of the *xyz*-index array under Initial Indices. The partition results are shown in Figure 4.26 under Initial Indices column in which the

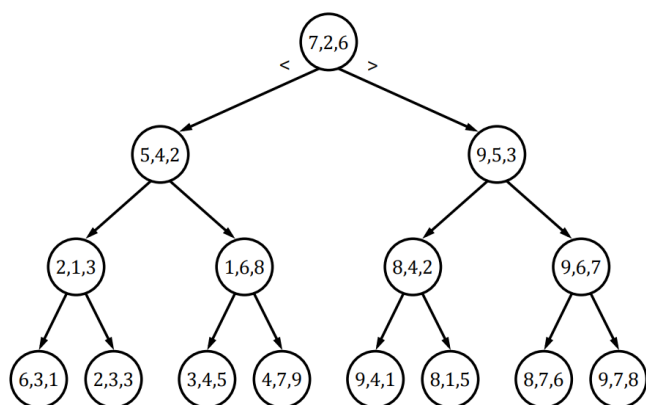
	PD Points (x,y,z)	Initial Indices xyz yzx zxy	After First Split xyz yzx zxy	After Second Split xyz yzx zxy
0	(2,3,3)	11 13 9	11 13 9	13 13 9
1	(5,4,2)	13 4 6	13 9 1	0 9 13
2	(9,6,7)	0 5 1	0 0 13	9 0 0
3	(4,7,9)	10 9 7	10 1 0	
4	(8,1,5)	3 0 13	3 10 10	11 10 10
5	(7,2,6)	1 6 0	1 11 11	10 11 11
6	(9,4,1)	9 1 12	9 3 3	3 3 3
7	(8,4,2)	5 7 10		
8	(9,7,8)	4 10 4	4 4 6	4 4 6
9	(6,3,1)	7 12 5	7 6 7	7 6 7
10	(3,4,5)	14 2 14	14 7 12	6 7 4
11	(1,6,8)	6 11 2	6 12 4	
12	(9,5,3)	12 14 11	12 2 14	14 2 14
13	(2,1,3)	2 8 8	2 14 2	2 14 2
14	(8,7,6)	8 3 3	8 8 8	8 8 8

Figure 4.26: Data slicing for 3-d tree

partitioning does not reorder the array of PD points. Instead, it reorders the yzx - and zxy -index arrays. Please note that xyz -index array requires no partitioning as it is already sorted in x direction and this was done when Initial Indices arrays were created. However, the yzx - and zxy -index arrays require partitioning in x direction by using the $x:y:z$ super key defined by median point 7:2:6. The partitioning of yzx index array is achieved as follows:

1. The elements of yzx -index array are compared to super key (median element of index array - 7:2:6)
2. They are copied either in upper half if they are less than the x value or in lower half if they are bigger than the x value from xyz super key.
3. The same procedure is repeated for zxy -index array.

The columns of After First Split reveal that the index value of 5 is absent from the index arrays since it represents the partitioning value. It also becomes the root of nascent K-d tree, as shown in Figure 4.27. The same procedure is also repeated for

Figure 4.27: A K-d tree built from (x, y, z) tuples

y direction, and the partitioned values (see column After Second Split in Figure 4.26) are removed and stored as children nodes of the root node. This recursive process is repeated until index array comprises of only one, two or three elements. In the case of only one point is left after the final split, it is automatically stored as a new node of K-d Tree. If there are two or three points left, these points are already sorted in the index array. So, the determination of which point referencing a new node and which point referencing from children is trivial.

4.3.3.6 Boost R Tree Algorithm

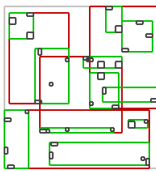
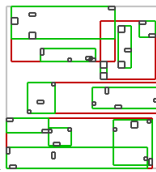
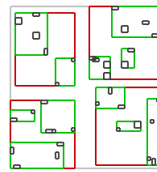
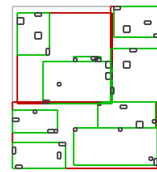
R-tree is currently the only spatial index implemented in Boost.Geometry.Index library which is a part of overall Boost library (Libraries (2009)). The intended use of Boost.Geometry.Index is to gather data structures defined as spatial indexes which can be used to accelerate searching for objects in multidimensional spaces. In general, spatial indexes store representations of geometric objects which allows the end user to search for objects occupying some space or object close to some point in a space.

R-tree is a tree data structure used for spatial queries and it is first proposed by Guttman (1984). Since all objects lie within a bounding rectangle, a query, that doesn't intersect the bounding rectangle, also cannot intersect any of the objects contained in the bounding rectangle. Similar to B-tree, R-tree is also a self-balanced search tree. The key part of balancing algorithm is the node splitting algorithm (Greene (1989)

and Beckmann et al. (1990)). Each algorithm produces different splits such that the internal structure of a tree may become different for each one of them. This means that more complex algorithms can better analyze the elements and produce less overlapping nodes. The tree with less overlapping nodes is more efficient in a search process because less nodes must be traversed in order to find desired objects. The downside of higher complexity algorithms is that analysis takes more time. In general, faster inserting results in slower querying and vice versa. Performance of R-tree is contingent on balancing algorithm, parameters and the data inserted into a container.

Most trees with searching algorithms (e.g. intersection, spatial search, nearest neighbor search) are rather simple. The key idea is to use the bounding boxes to decide whether or not to search inside a subtree. This means that most of the nodes in a tree are traversed during the search. R-trees are suitable for large data sets and databases, where the nodes can be paged to memory as needed and the whole tree cannot be kept in the main memory.

Table 4.1: Example structures of trees created by different algorithms and their operations times*

	Linear algorithm	Quadratic algorithm	R* -tree	Packing algorithm
Example structure				
1M Values inserts	1.76s	2.47s	6.19s	1.67s
100k spatial queries	2.21s	0.51s	0.12s	0.07s
100k knn queries	6.37s	2.09s	0.64s	0.52s

*(https://www.boost.org/doc/libs/1_55_0/libs/geometry/doc/html/geometry/spatial_indexes/introduction.html)

The main problem with an R-tree is that the rectangles do not encompass too much empty space and do not overlap too much (fewer subtrees need to be processed during the search). On the other hand, they are balanced (leaf nodes have the same height). Most of the research and improvements of R-trees are aimed at improving

the tree building process and they are defined by two main objectives; 1- Building an efficient tree from scratch (bulk-loading) and 2- Performing changes on an existing tree (insertion and deletion). Boost R-tree implements several building algorithms which are linear algorithm, quadratic algorithm, R*-tree algorithm and packing algorithm (bulk loading algorithm). As can be seen from **Table 4.1**, packing algorithm is faster when building the R-tree and also R-trees with better internal structure gives faster spatial and k nearest neighbors (*knn*) queries.

4.4 Solver methods

In previous sections initial steps of peridynamic analysis were presented. Geometry was defined and afterwards mesh was generated. Once that was done, family search was initiated and additional data was supplied such as material type and boundary conditions. Next step is to solve the problem. In VOPDSolver there are several solvers which are dependant on the type of peridynamic theory - Bond Based or Ordinary State Based. Furthermore solver type also depends on problem definition which can be dynamic, static or quasi-static.

In Figure 4.28 a simple flowchart for solver selection is described. In this flowchart it can be seen that first decision is dependant on the type of peridynamic theory. This is because there are significant differences between these two theories that should be dealt before proceeding forward, such as, peridynamic constants, surface correction factors, time step, mass vector, stiffness matrix and time loop structure. After selecting appropriate peridynamic theory, next step is to define if the problem at hand is of dynamic nature or static/quasi-static nature. This means that for dynamic problems, Explicit solver will be applied and for static/quasi-static problems, Direct or ADR solver will be used. Direct solver will be used for small to medium sized problems and ADR will be used for extremely large problems. In the following sections each of the solvers is explained in detail.

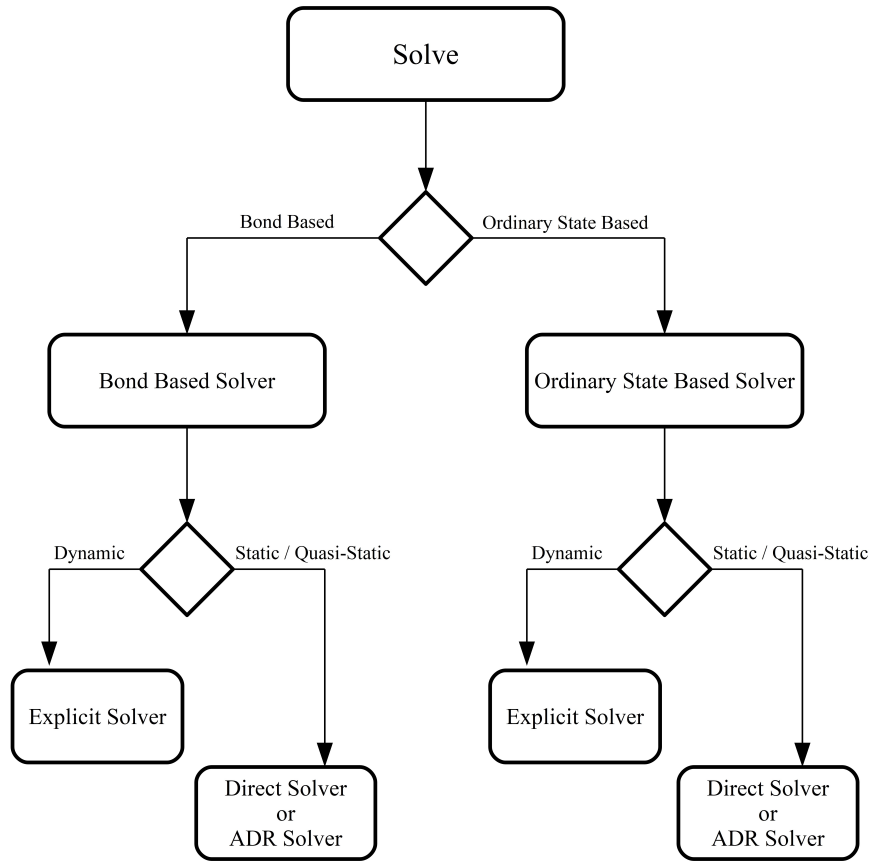


Figure 4.28: Solver selection flowchart

4.4.1 Dynamic Solver

By examining peridynamic equation of motion (Equation 3.18) it's obvious that the equation itself is in dynamic form, thus time integration can be performed by using explicit forward and backward difference technique (Silling (2004)). Peridynamic equation of motion for the explicit time integration scheme can be rewritten for the n^{th} time step as

$$\rho_k \ddot{\mathbf{u}}_k^n = \sum_j \left(\mathbf{t}_{kj}^n - \mathbf{t}_{jk}^n \right) \mathbf{V}_j + \mathbf{b}_k^n \quad \text{and} \quad n = 0, 1, 2, 3, \dots \quad (4.4)$$

where n is the time steps ($n = 0$ represents beginning of the analysis). After solving Equation 4.4 by summing force density vectors for each family member and finding

the unknown acceleration term $\ddot{\mathbf{u}}_k^n$, velocity and displacement of the material point k for the next time step can be determined by employing explicit forward and backward difference techniques. First, velocity is calculated with forward difference technique

$$\dot{\mathbf{u}}_k^{n+1} = \ddot{\mathbf{u}}_k^n \Delta t + \dot{\mathbf{u}}_k^n \quad (4.5)$$

Next step is to calculate displacement of the material point k at time step $n + 1$ by using backward difference technique

$$\mathbf{u}_k^{n+1} = \dot{\mathbf{u}}_k^{n+1} \Delta t + \mathbf{u}_k^n \quad (4.6)$$

The same procedure is repeated for every other material point. In explicit time integration, time step size must be very small otherwise the explicit scheme will become unstable. Stable time step size for explicit scheme was derived by Silling and Askari (2005) and further explained by Madenci and Oterkus (2014) as

$$\Delta t < \sqrt{\frac{2\rho_k}{\sum_j \left[2ad\delta \frac{(d\delta \sum_l \left(\frac{1}{|\xi_{lk}|} + \frac{1}{|\xi_{lj}|} \right) V_l)}{|\xi_{kj}|} + \frac{4b\delta}{|\xi_{kj}|} \right] V_j}} \cdot sFac} \quad (4.7)$$

Furthermore safety factor $sFac < 1$ is recommended in order to make the analysis more stable when dealing with nonlinearities in the structure.

4.4.2 Static and Quasi-static Solver

As it was mentioned in the previous Section 4.4.1 , peridynamic equation of motion is in dynamic form, but that doesn't mean it is not possible to solve static or quasi-static problems. There are two most common ways of solving static problems, one is solving peridynamic equation of motion by using Adaptive Dynamic Relaxation Underwood (1983). Other way is to rewrite equation of motion into a linear system of equations and solve them directly.

4.4.2.1 ADR - Adaptive Dynamic Relaxation

Dynamic relaxation (DR) is an explicit iterative method used for obtaining static solution of discretised continuum mechanics problem. There are several reasons behind popularity of DR method. First DR method is very useful for solving highly nonlinear problems (both geometric and material nonlinearities). Second, because the method is explicitly iterative there is no need for solving large system of equations. Furthermore, all quantities maybe treated as vectors, which will result in low memory composition when implementing DR into a computational code.

In order for DR to reach steady state solution an artificial mass dependent damping term is introduced in the equation of motion. This will in return reduce the oscillations in the transient response and by doing so converge towards the static solution.

$$\mathbf{M}\ddot{\mathbf{U}}(\mathbf{X}, t) + \mathbf{C}\dot{\mathbf{U}}(\mathbf{X}, t) + \mathbf{P}(\mathbf{U}(\mathbf{X}, t)) = \mathbf{F}(t) \quad (4.8)$$

where \mathbf{M} is artificial mass, \mathbf{C} is artificial damping and \mathbf{P} is a vector of internal forces. For linear problems \mathbf{P} is usually in the form of

$$\mathbf{P}(\mathbf{U}(\mathbf{X}, t)) = \mathbf{K}\mathbf{U}(\mathbf{X}, t) \quad (4.9)$$

where \mathbf{K} is a stiffness matrix and \mathbf{U} is vector of displacements. In case of dynamic relaxation, damping matrix \mathbf{C} has the form

$$\mathbf{C} = c\mathbf{M} \quad (4.10)$$

In Equation 4.10 c is the damping coefficient. In order to obtain the static solution of the equation of motion one needs to select appropriate damping coefficient c , time increment, and mass matrix \mathbf{M} such that

$$\mathbf{P}(\mathbf{U}) = \mathbf{F} \quad \text{or} \quad \mathbf{K}\mathbf{U} = \mathbf{F} \quad (4.11)$$

Unknown vector of displacements \mathbf{U} is determined by explicit time integration

scheme after certain number of iterations. In order not to overshoot the solution, which can be a large problem for certain problems (snap-trough buckling and plastic deformation) one needs to find the most effective damping coefficient to obtain the convergence. For this purpose Underwood (1983) introduced Adaptive Relaxation Method (ADR), where adaptive means that damping coefficient is determined at each time step. When applying ADR to peridynamic equation of motion Madenci and Oterkus (2014) rewrote the equation by adding new fictitious inertia and damping terms as follows

$$\mathbf{D}\ddot{\mathbf{U}}(\mathbf{X}, t) + c\mathbf{D}\dot{\mathbf{U}}(\mathbf{X}, t) = \mathbf{F}(\mathbf{U}, \mathbf{U}', \mathbf{X}, \mathbf{X}') \quad (4.12)$$

where \mathbf{D} is the fictitious density matrix and c is the damping coefficient. Values of \mathbf{D} and c are determined through application of the Greshgorin's theorem. Vector \mathbf{F} is made of peridynamic interactions and body forces, as

$$\mathbf{F}_k = \sum_{j=1}^N (\mathbf{t}_{kj} - \mathbf{t}_{jk})V_j + \mathbf{b}_k \quad (4.13)$$

For detailed explanation of peridynamic application of ADR methods reader can refer to Madenci and Oterkus (2014). Main advantage of ADR is the explicit nature that requires no need for solving large system of equations (direct solution). But if the problem at hand isn't too large, which in peridynamic sense means relatively coarse discretization and convergence of the static solution takes too many time steps, then direct solution could be a better option. Furthermore determining minimal number of iterations needed to reach steady state solution can be a tedious process and also minimal number of iterations is problem specific variable - it can't be used for other problems.

4.4.2.2 Direct Solution

Dealing with static or quasi static loading assumes that there are no inertial forces, acceleration is equal to zero. In case of quasi-static loading there is some acceleration when applying the load but it is very small and can be neglected, which means that

there are dynamic effects but they are extremely small and the system can be assumed to be static. In peridynamic terms this means acceleration $\ddot{\mathbf{u}}$ can be omitted from equation of motion. In case of bond based peridynamic theory, equation of motions can be formulated as

$$\sum_j \mathbf{f}_{kj} V_j + \mathbf{b}_k = 0 \quad (4.14)$$

In order to solve Equation 4.14 for every point in the body, it needs to be rewritten in matrix form. This means that the peridynamic force function needs to be expressed in terms of the second-order micromodulus tensor \mathbf{C} as

$$\mathbf{f} = \mathbf{C}(\xi)\eta \quad (4.15)$$

where

$$\mathbf{C}(\xi) = \frac{\partial \mathbf{f}}{\partial \eta}(0, \xi) \quad (4.16)$$

after solving Equation 4.16 and substituting \mathbf{C} in Equation 4.15 leads to matrix form of peridynamic equation of motion

$$\sum_j \mathbf{C}_{kj} \mathbf{u}_{kj} + \mathbf{b}_k = 0 \quad (4.17)$$

where $\mathbf{u} = [u_x, u_y, u_z, u'_x, u'_y, u'_z]^T$, $\mathbf{b} = [b_x, b_y, b_z]^T$ and \mathbf{C} is a local stiffness matrix for a single bond (for 3 DOF it has a size of 3x6). For an in-depth explanation of above procedure reader can refer to Silling (2000) and Bobaru et al. (2009). After summation over the entire body, standard equation of motion in matrix form is obtained

$$\mathbf{K}\mathbf{U} = \mathbf{B} \quad (4.18)$$

in which \mathbf{K} , \mathbf{U} and \mathbf{B} are the global stiffness, the displacement and the body load matrices of a body. Solving the Equation 4.18 is now a straightforward process by determining the inverse of stiffness matrix \mathbf{K}

$$\mathbf{U} = \mathbf{K}^{-1}\mathbf{B} \quad (4.19)$$

Although finding \mathbf{K}^{-1} looks like an easy job, it can become an exhausting process when dealing with complex structures. To this end there are well-known and effective numerical methods for finding an inverse of a matrix, such as

1. Gaussian elimination
2. LU decomposition
3. Cholesky decomposition
4. QR decomposition
5. Bruhat decomposition

Problem with above numerical methods is that in order for them to work in most optimal way one needs to spend substantial development time. On the other hand there are many third party libraries that do this exact job very efficiently. VOPDSolver is currently using Eigen C++ library for solving inverse problems. Furthermore according to Bobaru et al. (2009) an ill-conditioned stiffness matrix may be created due to the horizon size and the number of material points which can impede the process. Another problem that is especially apparent in peridynamics is the density of the stiffness matrix. In peridynamics stiffness matrix \mathbf{K} isn't as sparse as it is for FEM problems. This is because each row of stiffness matrix representing one material point has values for every DOF of every family point. This becomes even more obvious when number of family points increases due to larger horizon and/or going from 2D to 3D case.

4.4.2.2.1 Eigen C++ Library Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms. It supports all matrix sizes, from small fixed-size matrices to arbitrarily large dense matrices, and even sparse matrices. Furthermore it supports various matrix decompositions and geometry features. Eigen is very fast because fixed-size matrices are fully optimized: dynamic memory allocation is avoided, and the loops are unrolled when that makes sense and for large matrices, special attention is paid to cache-friendliness.

```

#include <Eigen\Sparse>
#include <Eigen/SparseLU>
VectorXd U(n), B(n);
SparseMatrix<double> K;
SparseLU<SparseMatrix<double>, COLAMDOrdering<int> > solver;
// fill K and U;
// Compute the ordering permutation vector from the structural pattern
  of K
solver.analyzePattern(K);
// Compute the numerical factorization
solver.factorize(K);
//Use the factors to solve the linear system
U = solver.solve(B);

```

Figure 4.29: Using Eigen C++ library to solve linear system of equations

In VOPDSolver Eigen SparseLU solver is currently used because it is optimized for small and large problems with irregular sparsity patterns which suits peridynamics. Figure 4.29 shows code snippet that demonstrates how to solve a problem $\mathbf{KU} = \mathbf{B}$ by using Eigen C++ library.

4.4.3 Parallel Execution

In order to speed up the codes researchers have long been using parallelization techniques. Programs can run parallel on standard desktop machines or in high performance computing (HPC) facilities. Furthermore parallelization can be done by using either central processing units (CPUs) or graphical processing units (GPUs). CPU and GPU parallelization can also be combined in order to solve large and complicated problems.

Any peridynamic code can be parallelized as long as it's solution method doesn't include matrices. In case of VOPDSolver that means Explicit and ADR solvers are parallelized. To be more exact SPMD (single program, multiple data) technique is used. In SPMD tasks are split up and run simultaneously on multiple processors with different input in order to obtain results faster. This means that in SPMD multiple autonomous processors simultaneously execute the same program at independent points. Part of the code that is parallelized is the calculation and summation of force density vector

```

#include <omp.h>
//find number of processors
SYSTEM_INFO sysinfo;
GetSystemInfo( &sysinfo );
int numCPU = sysinfo.dwNumberOfProcessors;

//split the loop into numCP-1 threads
#pragma omp parallel num_threads(numCPU-1)
{
    //set thread priority
    SetThreadPriority(GetCurrentThread(), THREAD_PRIORITY_HIGHEST);
    #pragma omp for
    for(int iNode = 0; iNode < totalNumOfPoints; iNode++)
    {
        //run the code for all material points (totalNumOfPoints)
    }
}

```

Figure 4.30: Example of for loop parallelization with openMP

for each material point.

In VOPDSolver only CPU parallelization is used by implementing openMP (Open Multi-Processing). OpenMP is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C++ on many platforms, instruction set architectures and operating systems. Figure 4.30 shows code snippet that demonstrates how to use openMP to parallelize for loop in the peridynamic code. It has to be mentioned that speed increase isn't linear. This means that if the computer has 4 processors with 2 threads each - 8 threads in total, speed increase want be 8 times. Code speedup test was done on the a Intel Core i7-3770 CPU @ 3.4GHz machine with 16 GB of internal memory which has 8 threads and the speedup was around factor 5.

4.5 Impact Definition

Impact modeling in peridynamics was first introduced by Silling and Askari (2004) where they presented multiple impact problems - Charpy V-notch test, accumulated damage in concrete due to multiple impacts, and crack fragmentation of a glass plate.

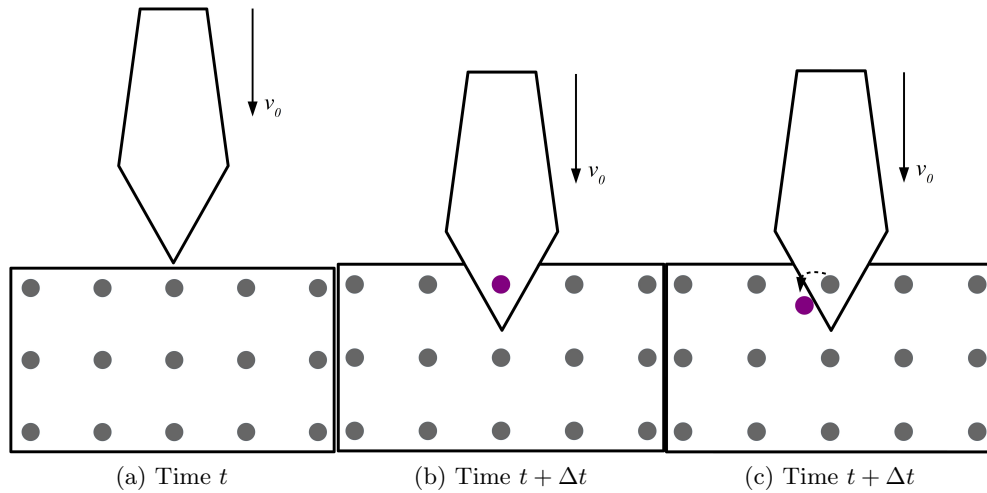


Figure 4.31: Rigid impactor and deformable body interaction (Madenci and Oterkus (2014))

Further work on impact modeling was done by Oterkus et al. (2012) where they presented impact damage of a reinforced panel under compression after impact due to a rigid penetrator. Theoretical overview of damage modeling framework was presented by Madenci and Oterkus (2014) where the impact can be modeled depending on the type of impactor. Impactors can be either flexible or rigid, where rigid impactors are not deformable at any instant and flexible impactors are deformable and governed by the peridynamic equation of motion.

Rigid impactor framework is currently implemented in VOPDSolver and will be explained in the rest of this section. In rigid framework target body is deformable and governed by peridynamic equation of motion while impactor is rigid. At first sign of contact between impactor and target there will be an initial penetration of target body into the impactor and vice versa. For physical reality to be satisfied material points from the target body need to be expelled just outside of the impactor. New location of the material point is always perpendicular to the closest surface of the impactor see Figure 4.31.

Once the point \mathbf{x}_k is expelled outside the impactor new velocity in its new location

at the next time step, $t + \Delta t$, can be computed as

$$\bar{\mathbf{v}}_k^{t+\Delta t} = \frac{\bar{\mathbf{u}}_k^{t+\Delta t} - \mathbf{u}_k^t}{\Delta t} \quad (4.20)$$

where $\bar{\mathbf{u}}_k^{t+\Delta t}$ is new displacement vector due to change in location at time step $t + \Delta t$, with Δt being time increment value. Displacement before contact at time step t is \mathbf{u}_k^t . Force experienced by the impactor at $t + \Delta t$ exerted from material point \mathbf{x}_k can be computed as

$$\mathbf{F}_k^{t+\Delta t} = -1 \times \rho_k \frac{\bar{\mathbf{v}}_k^{t+\Delta t} - \mathbf{v}_k^{t+\Delta t}}{\Delta t} V_k \quad (4.21)$$

where $\mathbf{v}_k^{t+\Delta t}$ is velocity before relocating the material point \mathbf{x}_k at time step $t + \Delta t$, with ρ_k and V_k being density and volume. In order to calculate overall impact force on the rigid impactor it is necessary to sum all of the contributions from every material points as

$$\mathbf{F}^{t+\Delta t} = \sum_k \mathbf{F}_k^{t+\Delta t} \lambda_k^{t+\Delta t} \quad (4.22)$$

where

$$\lambda_k^{t+\Delta t} = \begin{cases} 1 & \text{inside impactor} \\ 0 & \text{outside impactor} \end{cases} \quad (4.23)$$

4.5.0.1 Impact examples

In this section several examples are presented. First two problems are validation examples from Madenci and Oterkus (2014), where impact of two identical flexible bars is 3D problem and rigid disk impacting on a rectangular plate is a 2D example. Both examples are compared to FEA results using ANSYS. Third problem is lighthouse example from Bjerkas et al. (2009).

Impact of Two Identical Flexible Bars

Detailed results and problem description can be found in Madenci and Oterkus (2014), following is just the basic problem description:

Geometric Parameters

Length of the identical bars: $L = 0.05$ m

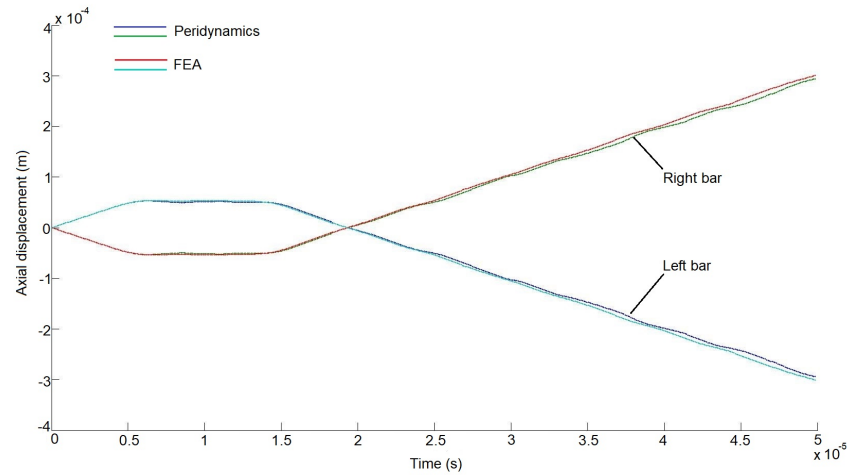


Figure 4.32: Impact of two identical flexible bars - displacement predictions in the x-direction at the centers $(\pm 0.025, 0.0, 0.0)$ of the left and right bars as time progresses for peridynamics and FEA

Width of the identical bars: $W = 0.01$ m

Thickness of the identical bars: $h = 0.01$ m

Material Properties

Youngs modulus: $E = 75$ GPa

Poissons ratio: $\nu = 0.25$

Mass density: $\rho = 2700$ kg/m³

Initial Conditions

Initial condition of the bars: $\dot{u}_x = \pm 10$ m/s

As it can be seen from the Figure 4.32 peridynamic results are in good agreement with FEA analysis. Furthermore, if compared to Figure 10.3 from Madenci and Oterkus (2014) it can be seen that results are identical.

A Rigid Disk Impacting on a Rectangular Plate

Detailed results and problem description can be found in Madenci and Oterkus (2014), following is just the basic problem description:

Geometric Parameters

Length of the identical bars: $L = 0.2$ m

Width of the identical bars: $W = 0.1$ m

Thickness of the identical bars: $h = 0.009$ m

Material Properties

Youngs modulus: $E = 191$ GPa

Poissons ratio: $\nu = 1/3$

Mass density: $\rho = 8000$ kg/m³

Impactor Properties

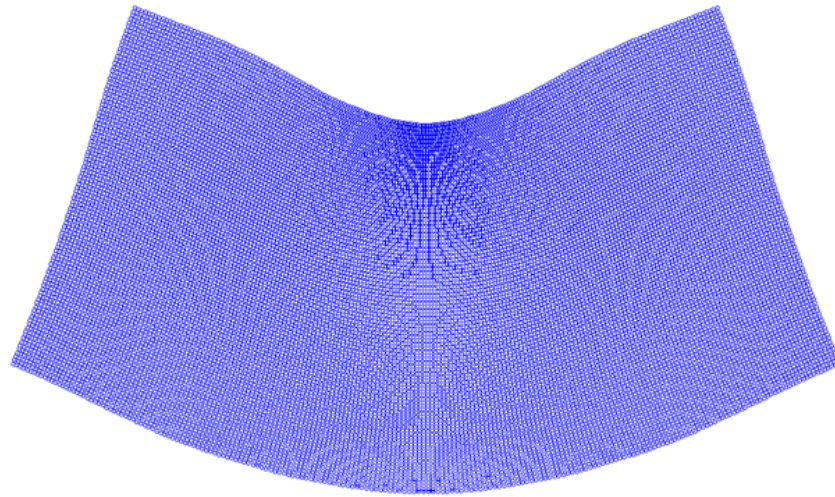
Diameter of the impactor: $D = 0.05$ m

Thickness of the impactor: $H = 0.009$ m

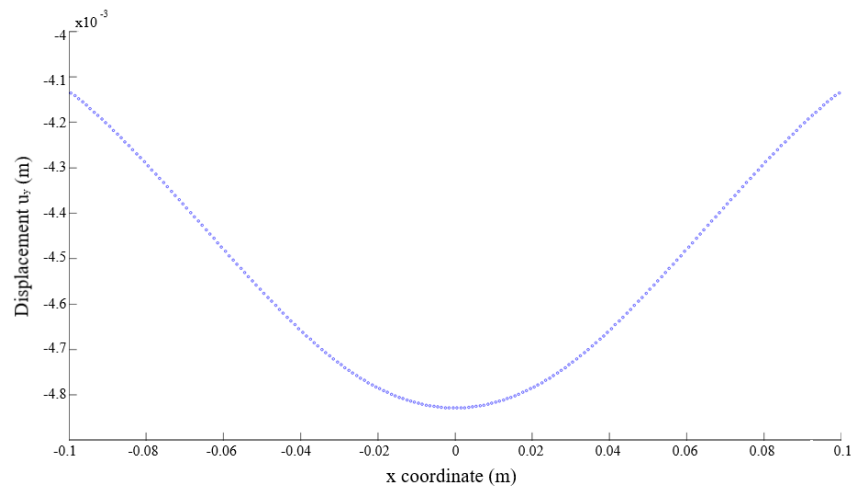
Initial velocity of the impactor: $v = 32$ m/s

Mass of the impactor: $m = 1.57$ kg

In Figure 4.33a, deformed peridynamic model is shown which has a good agreement with physical reality. Furthermore, comparing Figure 4.33b to Figure 10.7a from Madenci and Oterkus (2014) a good agreement between both peridynamic results and FEA analysis can be observed.



(a) Deformed peridynamic model



(b) Peridynamic displacement prediction in the y-direction at a time step of 2000 along the central x-axis

Figure 4.33: A rigid disk impacting on a rectangular plate peridynamic model

Chapter 5

Examples

5.1 Introduction

This chapter contains a detailed overview of problems that were examined in order to show that peridynamic theory and VOPDSolver that was subsequently developed can be utilised to solve complex fracture and impact problems. Most of these examples will deal with ice structure interactions which are the reason behind this PhD thesis. Furthermore certain number of problems will be examined in order to give answers related to reduction of computational burden due to complexities of ice structure interactions and peridynamics itself.

With this in mind first example in this chapter will be *Dynamic propagation of a macrocrack interacting with parallel small cracks*. The reason behind this study was to show that peridynamics can be used to solve complex fracture problems and that VOPDSolver can model those problems with ease. This was important for future research on ice structure interactions which are both hard to model and complicated to solve. In this study, the effect of small cracks on the dynamic propagation of a macrocrack is investigated by using a new continuum mechanics formulation, peridynamics. Various combinations of small cracks with different number, location and density are considered. Depending on the location, density and number of small cracks, the propagation speed of macrocrack differs. Some combinations of small cracks slows down the propagation of a macrocrack by 34%. Presented results show that this analysis can be

useful for the design of new microstructurally toughened materials.

Next example is *Peridynamic Model for a Mindlin Plate Resting on A Winkler Elastic Foundation* which become a cornerstone for future static analysis of ice structure interactions. In this study, a peridynamic model is presented for a Mindlin plate resting on a Winkler elastic foundation. Winkler foundation was introduced in order to introduce simplified effects of an ice sheet floating on water. In this study it will be shown that, although Winkler foundation is relatively coarse simplification of fluid structure interaction it is still extremely capable in capturing realistic fracture behaviour. In order to achieve static and quasi-static loading conditions, direct solution of the peridynamic equations is utilised by directly assigning inertia terms to zero rather than using widely adapted adaptive dynamic relaxation approach. The formulation is verified by comparing against a finite element solution for transverse loading condition without considering damage and comparing against a previous study for pure bending of a Mindlin plate with a central crack made of polymethyl methacrylate material having negligibly small elastic foundation stiffness. Finally, the fracture behaviour of a pre-cracked Mindlin plate rested on a Winkler foundation subjected to transverse loading representing a floating ice floe interacting with sloping structures. Similar fracture patterns observed in field observations were successfully captured by peridynamics.

Following two examples are to a certain extent continuation of *Peridynamic Model for a Mindlin Plate Resting on A Winkler Elastic Foundation*. First study is *In-Plane and Out-Of Plane Failure of an Ice Sheet Using Peridynamics* where ice sheet failure modes are examined. This is important especially with designs for offshore structures/icebreakers or predicting ice cover's bearing capacity for transportation, where it is essential to determine the most important failure modes of ice. Structural properties, ice material properties, ice-structure interaction processes, and ice sheet geometries have significant effect on failure modes. In this paper two most frequently observed failure modes are studied; splitting failure mode for in-plane failure of finite ice sheet and out-of-plane failure of semi-infinite ice sheet. Peridynamic theory was used to determine the load necessary for in-plane failure of a finite ice sheet. Moreover, the relationship between radial crack initiation load and measured out-of-plane failure load

for a semi-infinite ice sheet is established. To achieve this, two peridynamic models are developed. First model is a 2 dimensional bond based peridynamic model of a plate with initial crack used for the in-plane case. Second model is based on a Mindlin plate resting on a Winkler elastic foundation formulation for out-of-plane case. Numerical results obtained using peridynamics are compared against experimental results and a good agreement between the two approaches is obtained confirming capability of peridynamics for predicting in-plane and out-of-plane failure of ice sheets. Second study is *Peridynamic Approach for Modelling Ice-Structure Interactions* where it is going to be shown that peridynamics can be used to establish correct failure pattern for ice structure interactions both in 2D and 3D. Furthermore this study will showcase that VOPDSolver is capable of modelling complex three dimensional impact interactions and solve large problems within reasonable time due to use of parallelization. Having said that, ice-structure interaction modelling is still a very challenging process since ice material response depends on variety of factors including applied load, strain rate, temperature, and salinity. One of the most significant and widely studied ice problems from mechanical standpoint is out-of-plane failure of an ice floe. Out-of-plane failure process is very important since it occurs in different applications including ice covers bearing capacity for transportation, and bending failure of level ice interacting with a sloping structure. Therefore, it is important to perform a detailed structural analysis of ice fracture response under out of plane bending conditions in order to better understand its behaviour. As a new continuum mechanics formulation, peridynamics, can be very useful when dealing with fractures due to its various advantages with respect to some other traditional techniques including linear elastic fracture mechanics, cohesive zone model and extended finite element method. Hence, in this study, peridynamic analysis of ice plate under out of plane bending load for 2D and 3D cases is studied. Both models are qualitative in nature as only fracture patterns were observed. For the 2D analysis, peridynamic Mindlin plate on the elastic foundation was developed in order to study ice behavior under out of plane loading with the respect to the ice flow size. By changing the size of the plate, crack propagation patterns should be influenced, which will in return allow us to see if finite ice flow of a certain size can represent

semi-infinite ice flow. For the 3D example a standard peridynamic structural model with rudimentary buoyancy body load representing the fluid base which is interacting with a conical rigid body, was used, in order to recreate real life loading conditions (ice breaker going through the ice) and subsequently produce fracture patterns that can be seen in the literature.

Next study is *Ice-structure Interaction Applied to the Lighthouse Example* where interaction between vertical cylindric structures and ice sheet is examined. Floating ice sheets interacting with rigid vertical structures such as oil platforms represent a significant problem when designing arctic offshore structures. These structures regularly experience high ice loads for prolonged periods of time since ice sheets fail by crushing. Currently estimation of ice loads on vertical structures is still done by utilizing empirical methods derived from case studies in model-scale or by measuring full-scale ice loading events. As with all estimation methods there is some doubt as to how applicable are those methods when we have varying boundary conditions, ice properties, ice sheet thickness or drift speed. This means that even today, most reliable way of testing a new design is to do a model-scale test to verify the exerted ice loads to structures and that is a time consuming task. With this in mind peridynamic was used to estimate ice loads and predict failure patters. In this study we will show that simple 2D bond based peridynamic model is capable not only to predict reasonable loads but also show realistic fracture behaviour of an ice sheet.

Last example in this PhD thesis will be *Family Member Search Algorithms for Peridynamic Analysis* where different family search algorithms that are currently implemented in VOPDSolver will be examined. With that being said, family search process is one of the most time-consuming parts of a peridynamic analysis. Especially for problems which require continuous update of family members inside the horizon of a material point, the time spent to search for family members becomes crucial. Hence, efficient algorithms are required to reduce the computational time. In this study, various family member search algorithms suitable for peridynamic simulations are presented including brute-force search, region partitioning, and tree data structures. By considering problem cases for different number of material points, computational time between

different algorithms is compared and the most efficient algorithm is determined.

5.2 Dynamic propagation of a macrocrack interacting with parallel small cracks

5.2.1 Introduction

Small cracks can have significant effect on the dynamic propagation behaviour of macrocracks if they are within the range of the macrocrack's influence domain. The location of small cracks and their orientation with respect to the tip of a macrocrack are important, since the stress intensity of the macrocrack tip changes accordingly. Numerical tools can be useful to investigate such an interesting and important problem.

Numerical prediction of crack growth in computational mechanics has been and it still is a considerable problem that can't be easily solved by conventional numerical methods such as cohesive elements (Hillerborg et al. (1976), Xu and Needleman (1994)) and extended finite element method (XFEM) (Sukumar et al. (2000), Moës and Belytschko (2002)). XFEM require damage criterion and careful stress tracking around the crack tip in order to decide if crack is going to branch or not. On the other hand, mesh dependency is a problem for cohesive elements. All of the above problems are making it difficult to correctly simulate crack propagation and especially multiple crack propagation/interaction (Ha and Bobaru (2010)). Meshfree methods can be a good alternative to finite element method and has been applied to dynamic fracture and fragmentation (Benz and Asphaug (1995), Rabczuk and Belytschko (2007)). Moreover, Cracking Particle Method (CPM) (Rabczuk and Belytschko (2004)) was introduced for complex fracture patterns such as crack branching and coalescence. Another promising approach for fracture modelling is lattice methods which represent a medium as a connection of interacting nodes or particles (Griffiths and Mustoe (2001), O'Brien and Bean (2011), Pazdniakou and Adler (2012), Morrison et al. (2015)).

In this study, it is shown that using a new continuum mechanics formulation, peridynamics, as an alternative method, it is possible to correctly model and simulate dynamic fracture, in particular multiple small cracks interacting with a macrocrack in

brittle materials. Peridynamics (Silling (2000), Silling and Askari (2005), Silling et al. (2007)) doesn't need failure criteria for crack propagation as in XFEM or cohesive FEM methods, it is something that is contained within the peridynamic methodology and is achieved through a simple bond based failure criterion that is derived from material's energy release rate. Moreover, multiple cracks can easily be analysed since cracks are not treated as special objects in the formulation. Peridynamics has been successfully used to analyze different material systems and geometrical configurations (Diyaroglu et al. (2015), Oterkus and Madenci (2015), Perré et al. (2016), Oterkus and Madenci (2012), Oterkus et al. (2010)). An extensive literature survey on peridynamics is given in Madenci and Oterkus (2014).

5.2.2 Problem Definition

A rectangular plate with dimensions of 0.05 m by 0.05 m is considered as shown in Figure 5.1. Material is chosen as PMMA. Plate contains a macrocrack that interacts with multiple small cracks. In the case of bond based peridynamics, there is a constraint on Poisson's ratio as $1/3$ for 2-Dimensional problems due to the assumption of pairwise forces between material points which is slightly lower than the actual Poisson's ratio of PMMA. However, for dynamic fracture problems, Poisson's ratio can have insignificant influence on speed and direction of crack propagation, Ha and Bobaru (2010).

In this study, the plate is considered under tension loading and it is subjected to a high velocity boundary condition of 5 m/s. Several types of small crack configurations are considered; single small crack collinear to the main crack, two symmetrical small cracks, horizontal and transverse array of small cracks, to investigate the effect of small cracks on the macrocrack behaviour by calculating the crack propagation speed.

5.2.3 Numerical Results

Peridynamic model used for this study is defined with a fixed horizon size of $\delta = 3\Delta x$, where Δx is the spacing between material points and it is specified as 0.0001 m. Therefore, a total number of 250000 material points exist in the model. Boundary region is equal to horizon size, δ and thickness of the plate is specified as $h = \Delta x$ (Figure

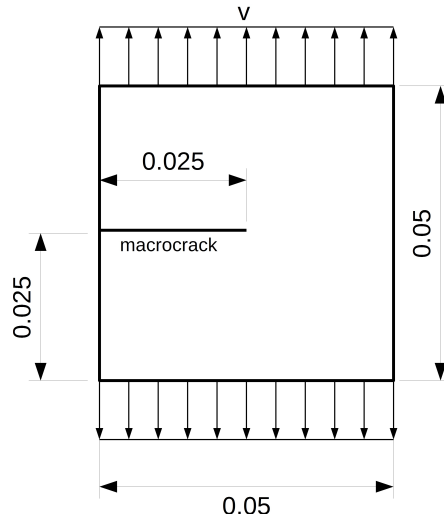


Figure 5.1: Problem definition

5.2). Critical stretch is set to $s_0 = 0.030857$. The time step is chosen as $\Delta t = 4 \times 10^{-8}$ s and the number of time steps is 2000.

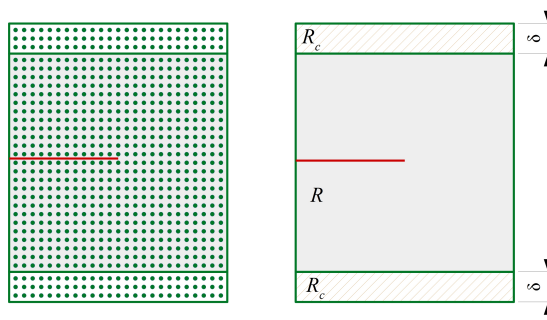


Figure 5.2: Peridynamic model and its discretization

Finally, the length of a small crack is defined as

$$l_{smallcrack} \ll L_{macrocrack} \rightarrow l_{smallcrack} = l_{macrocrack}/50 \quad (5.1)$$

5.2.3.1 Macrocrack Propagation

In the first example, the crack propagation of a macrocrack is investigated without considering small cracks in the model in order to compare with those cases including small cracks. It is observed that the macrocrack started to propagate around a time

step of 500 and reached the end of the plate after 1700 time steps. The crack speed is calculated by comparing the crack length at 4×10^{-5} s with the initial crack length (Figure 5.3) as

$$v_{cp} = \frac{\delta l}{dt} = 480 \text{ m/s} \quad (5.2)$$

5.2.3.2 One Small Crack Collinear to a Macrocrack

In the second example, a single small crack is aligned with the macrocrack as shown in Figure 5.4 (Wang et al. (2016), Rubinstein (1985)).

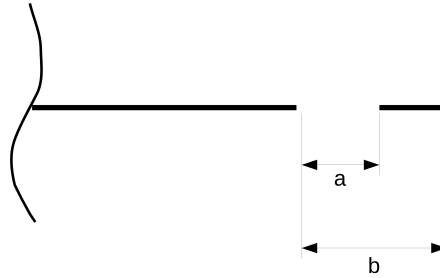


Figure 5.4: Collinear small crack in front of a macrocrack

The macrocrack propagation speed is calculated for different ratios of a/b as given in Table 5.1. The minimum crack propagation speed is obtained for the highest a/b ratio which corresponds to furthest small crack. By comparing with the only macrocrack case, the collinear small crack causes an increase in macrocrack propagation speed. However, no effect on initiation time is observed, i.e., all the cases have the same initiation time as benchmark case - 500 time steps. The effect of a/b ratios on the crack shapes can be seen in Figure 5.5.

Table 5.1: Macrocrack propagation speed at 4×10^{-5} s

	$a/b = 0.2$	$a/b = 0.4$	$a/b = 0.6$	$a/b = 0.8$
v_{cp} (m/s)	562.5	562.5	557.5	515

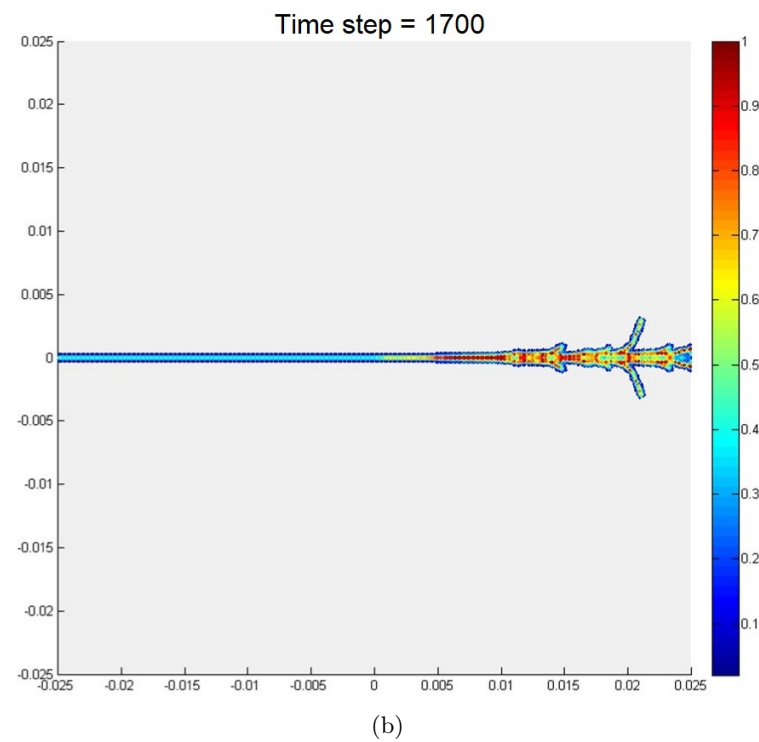
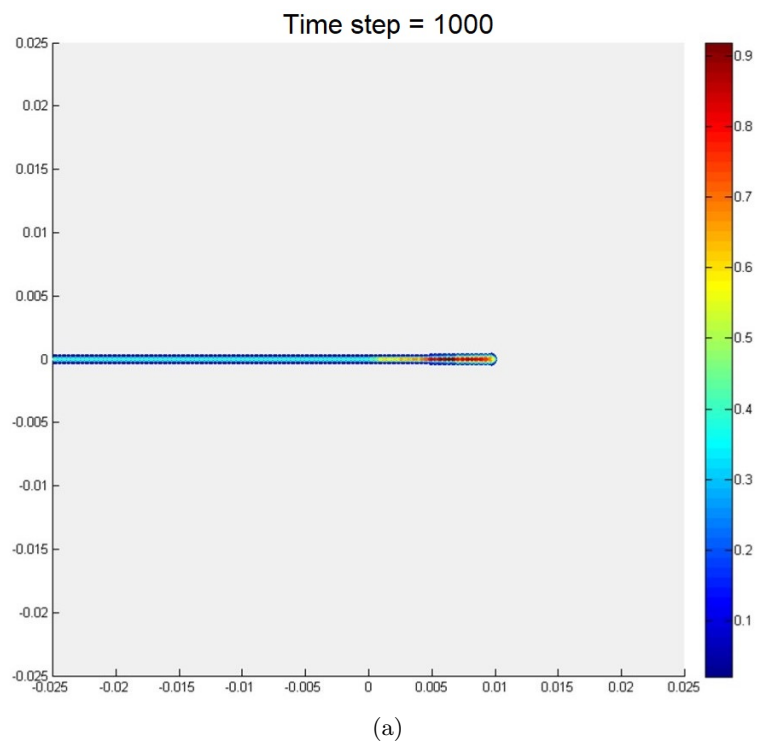


Figure 5.3: Macrocrack propagation without considering small cracks, (a) 4×10^{-5} s, (b) 6.8×10^{-5} s

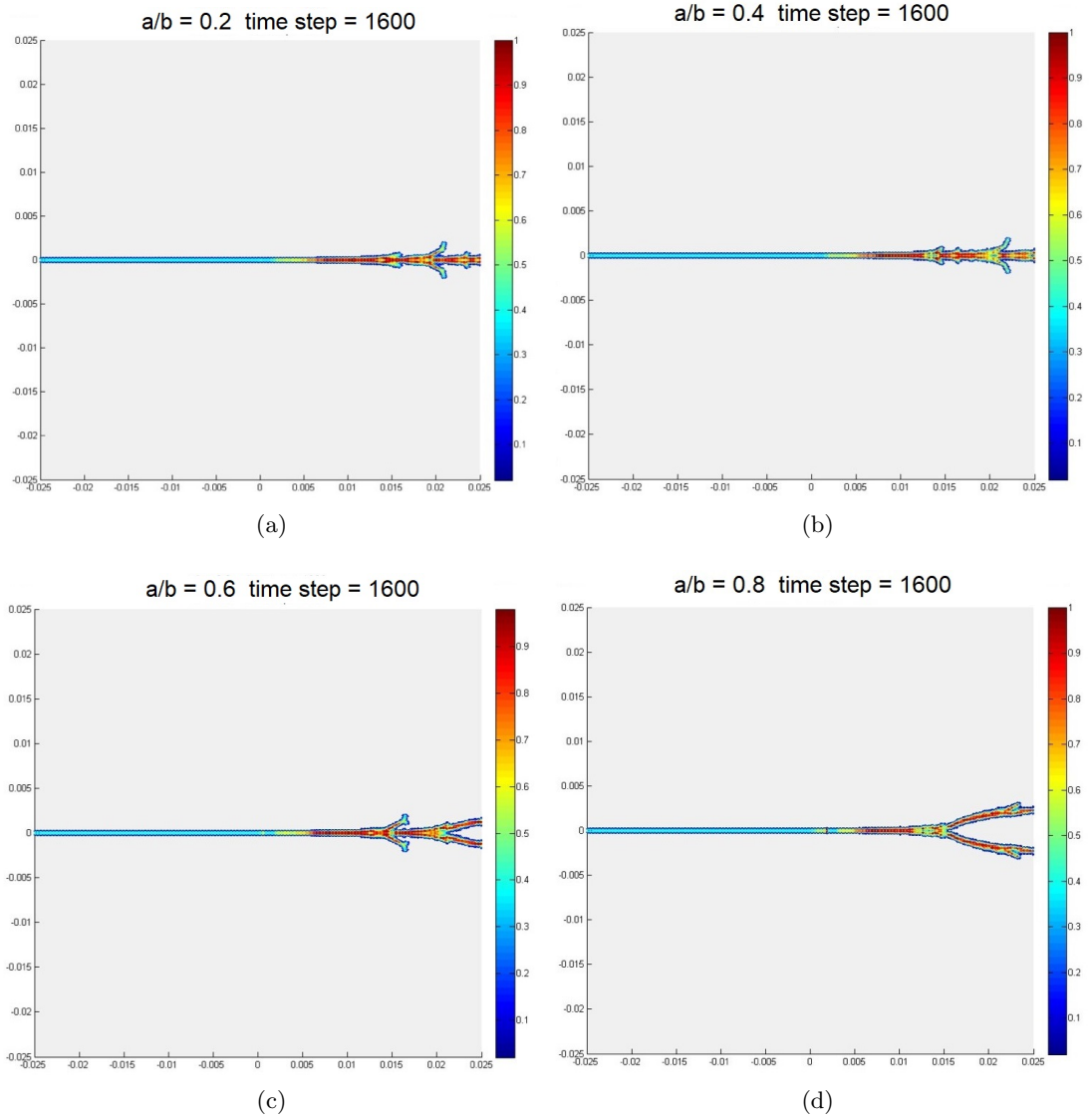


Figure 5.5: Macrocrack propagation for different values of a/b (a) $a/b = 0.2$, (b) $a/b = 0.4$, (c) $a/b = 0.6$, (d) $a/b = 0.8$ at 6.4×10^{-5} s

5.2.3.3 Two Parallel Small Cracks

In the third example, two symmetric small cracks are positioned on both sides of the macrocrack (Figure 5.6).

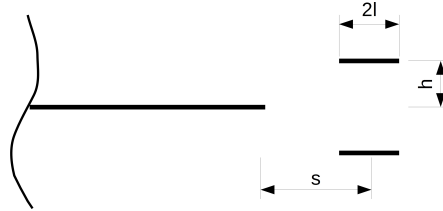


Figure 5.6: Two parallel small cracks

The macrocrack propagation speed is calculated for varying h/l and s/l values. It is found that for $s/l = 2$, the influence of the small cracks on the speed of macrocrack propagation is insignificant and the shape of the crack path is very similar to that of a single macrocrack (Figure 5.7).

Moreover, h/l ratio has the least influence on the overall macrocrack propagation as shown in Table 5.2. The minimum crack speed is obtained for $h/l = 1.25$ and $s/l = 0$ (Figure 5.8). It is worth noting that for $s/l = 0$ and $s/l = 2$, the crack propagation is observed in small cracks only. This behaviour is also observed by Wang et al. (2016). They observed that for $h/l = 0.75$ and 1, the influence of the stress intensity factor of small cracks is greater than the macrocrack. A similar behaviour is also observed for $h/l = 1.25$

Table 5.2: Macrocrack propagation speed at 4×10^{-5} s

	$s/l = -2$	$s/l = 0$	$s/l = 2$
$h/l = 0.75$	480 m/s	356.25 m/s	356.25 m/s
$h/l = 1$	480 m/s	356.25 m/s	356.25 m/s
$h/l = 1.25$	480 m/s	356.25 m/s	356.25 m/s

5.2.3.4 Multiple Small Cracks Interacting with the Macrocrack

For the last example, interactions between macrocrack and multiple small cracks are investigated. First, a set of horizontal cracks (see Figure 5.9) for $h/l = 1$ and h/l

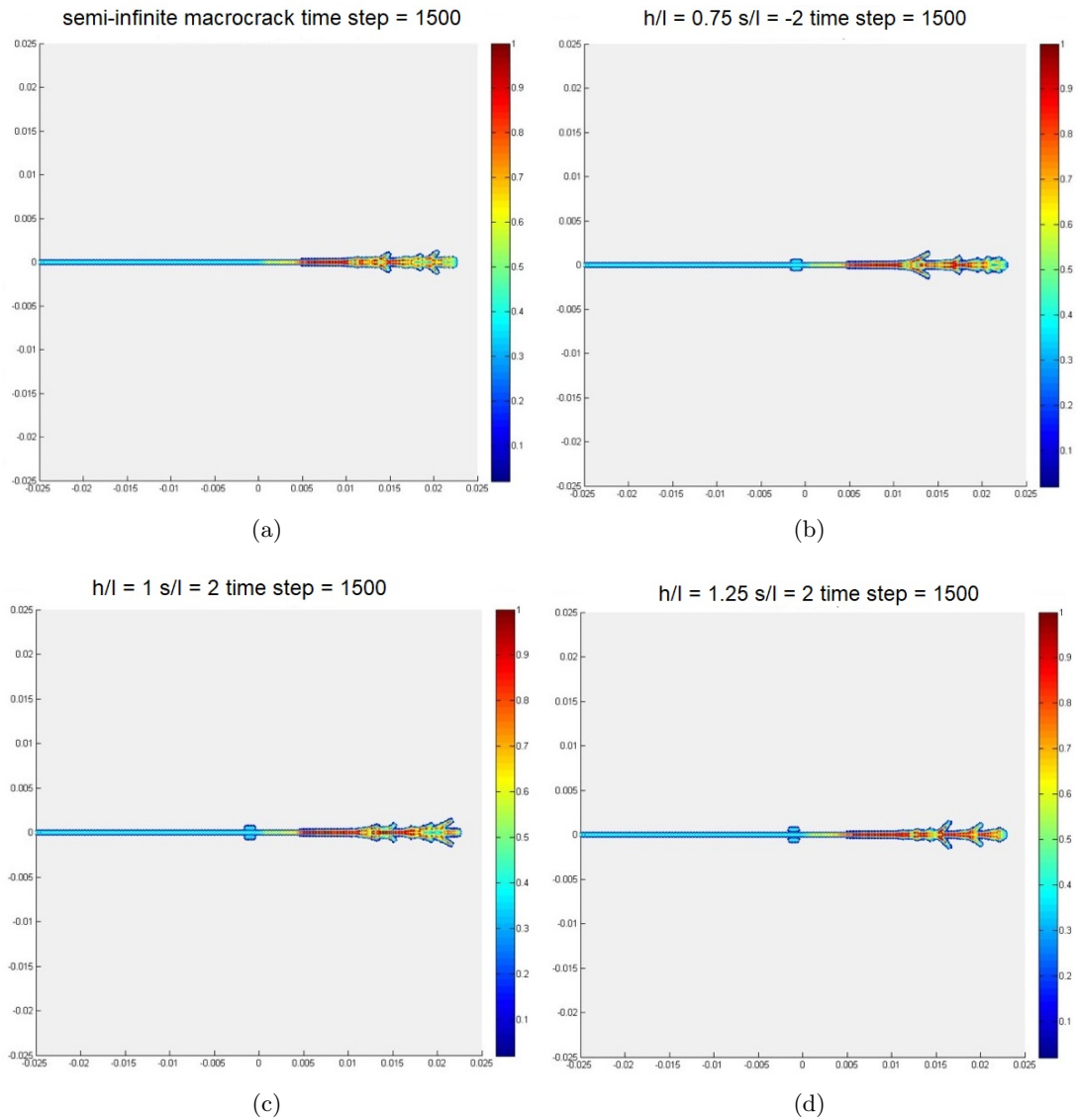


Figure 5.7: Comparison of a single macrocrack and macrocrack with two parallel small cracks (a) macrocrack without a small crack, (b) $h/l = 0.75$, $s/l = 2$, (c) $h/l = 1$, $s/l = 2$, (d) $h/l = 1.25$, $s/l = 2$ at 6×10^{-5} s

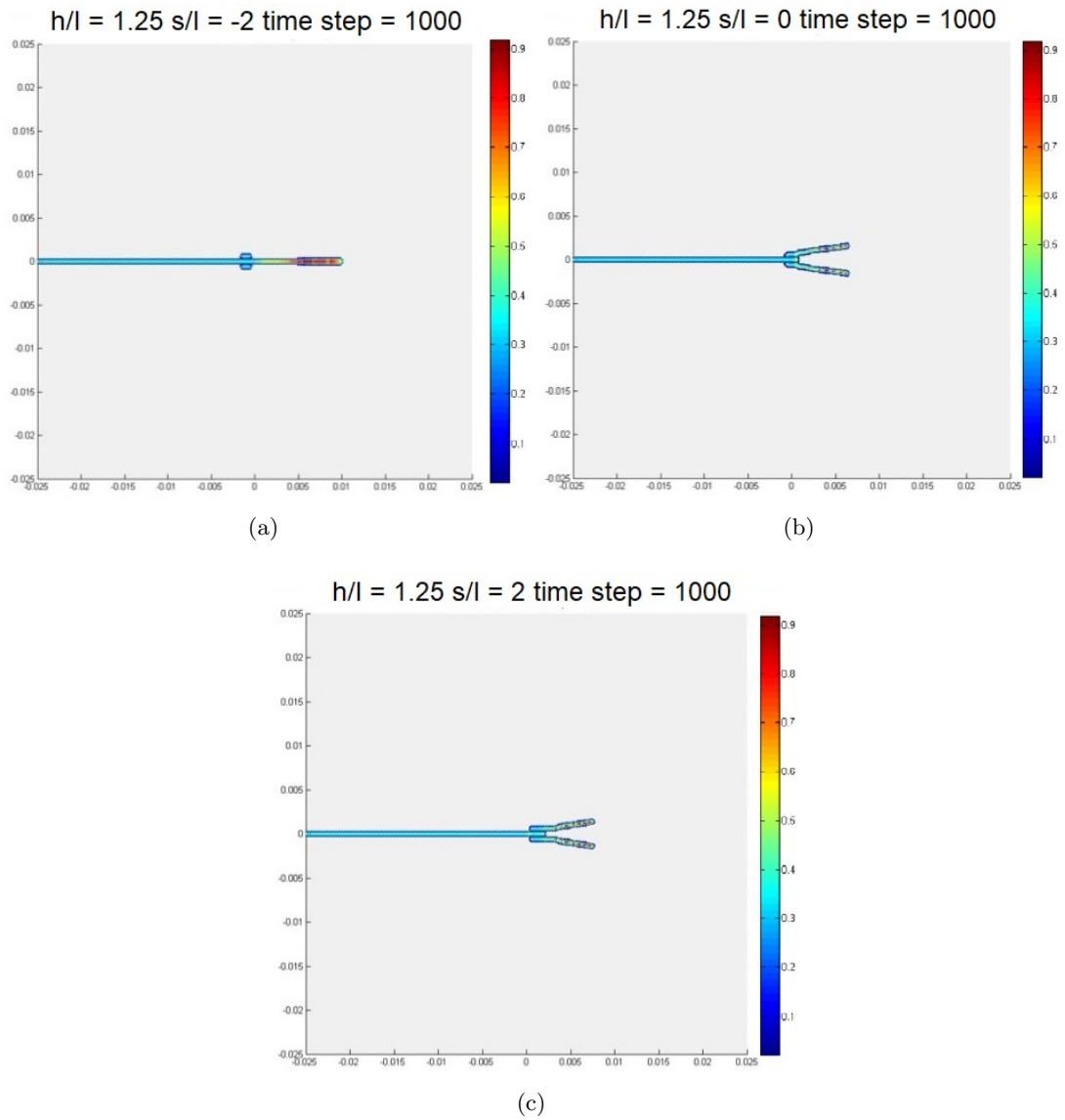


Figure 5.8: Crack propagation for $h/l = 1.25$ (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s

Table 5.3: Overall crack propagation length of horizontal small cracks at 4×10^{-5} s

	$s/l = -2$	$s/l = 0$	$s/l = 2$
x (m)	0.00975	0.00985	0.01075

$= 2.5$ are investigated. It is observed that H/l should be greater than 2 in order to exclude the effect between adjacent columns of small cracks (Wang et al. (2016)).

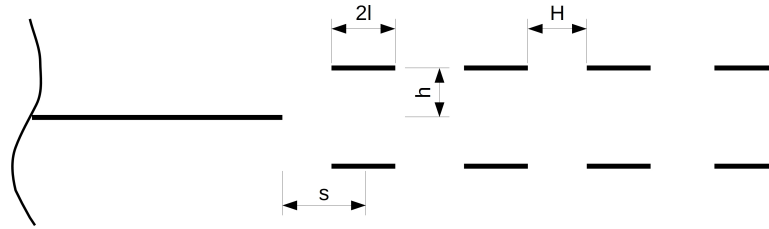


Figure 5.9: Set of horizontal small cracks

The crack propagation speed is calculated for different ratios of s/l as given in Table 5.3. It is found that the present small crack configuration stopped the macrocrack propagation. However, the present configuration has no influence on crack initiation time and a very low influence on the overall crack propagation length (Table 5.3 and Figure 5.10).

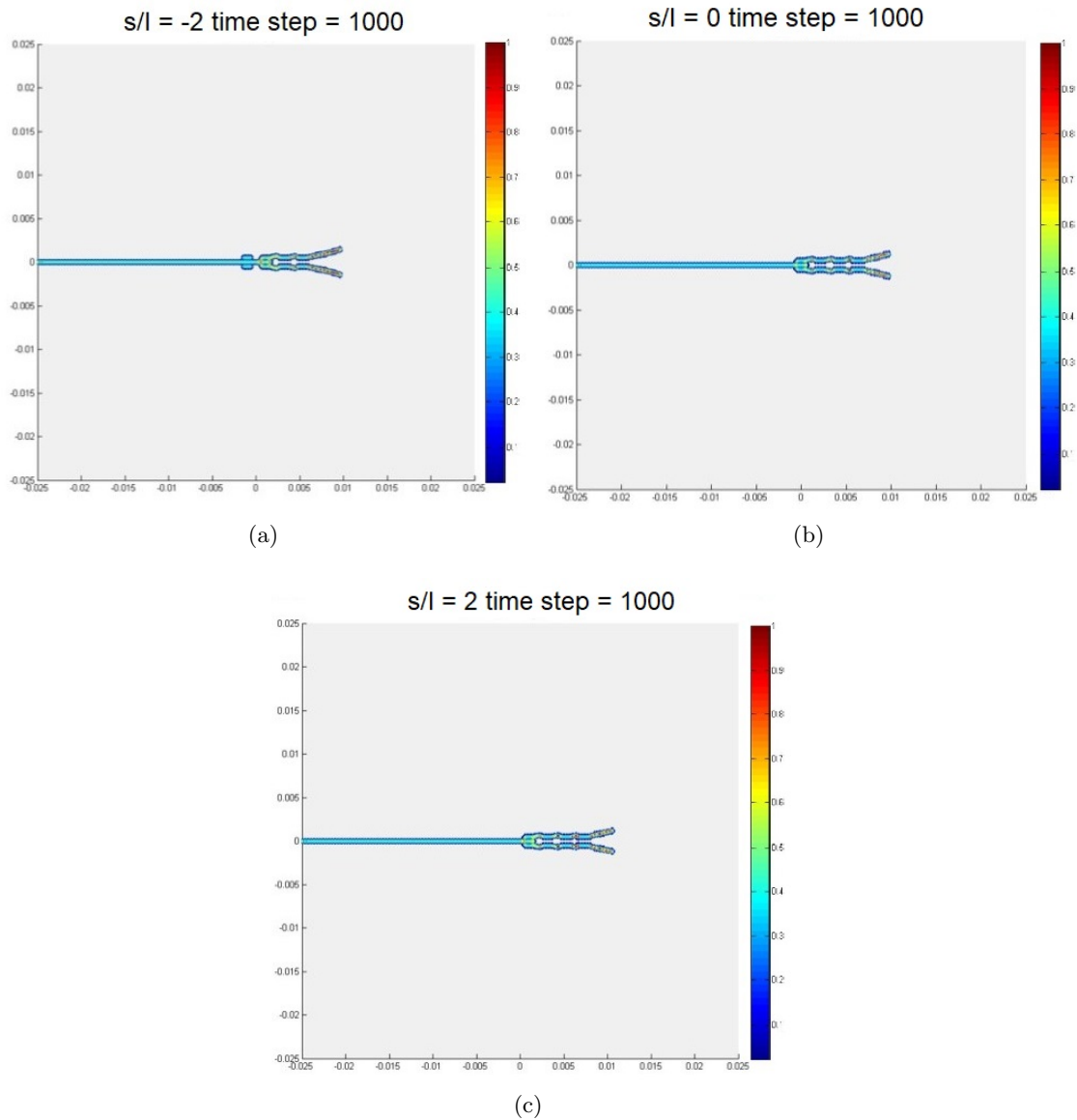


Figure 5.10: Crack propagation of horizontal small cracks (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s

Moreover, the effect of multiple small cracks in the vertical direction is investigated (Figure 5.11) for $h/l = 1$ and $s/l = 2, 0$ and 2 .

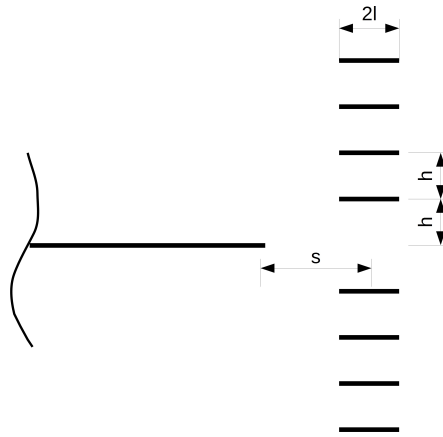


Figure 5.11: Set of vertical small cracks

The crack propagation speed is calculated for different ratios of s/l as given in Table 5.4. It is found that the crack propagation speed is reduced by 34% for $s/l = 0$ compared to a single macrocrack case (Table 5.4). The results show good correlation with conclusions from Wang et al. (2016) where they observed relatively high amplification decrease for $h/l = 1$ (Figure 5.12).

Table 5.4: Crack propagation speed of vertical small cracks at 4×10^{-5} s

	$s/l = -2$	$s/l = 0$	$s/l = 2$
v_{cp} (m/s)	480	316.66	333.33

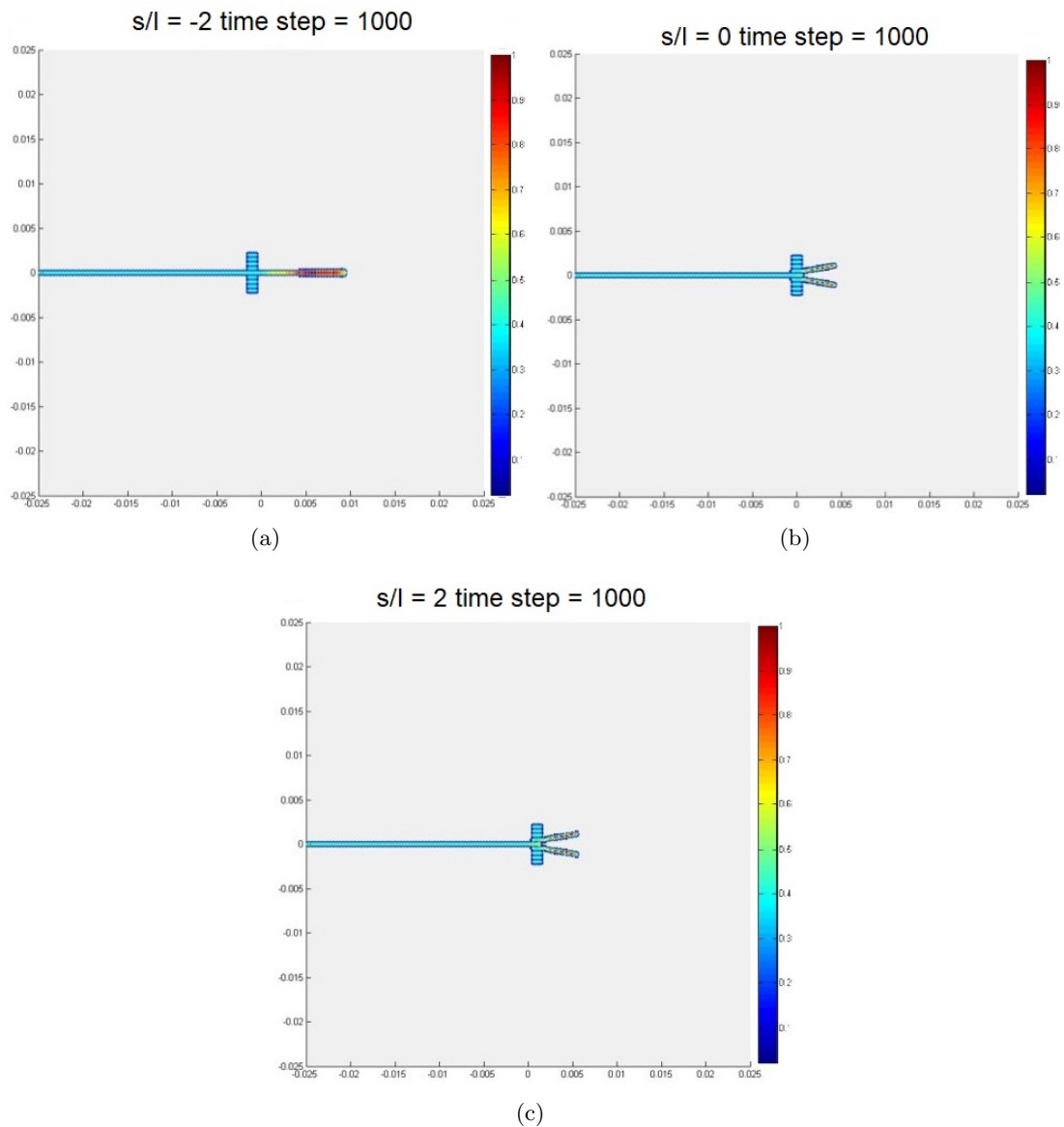


Figure 5.12: Crack propagation of vertical small cracks (a) $s/l = -2$, (b) $s/l = 0$ and (c) $s/l = 2$ at 4×10^{-5} s

Finally, the superior capability of peridynamics for fracture prediction is demonstrated by considering a significantly large number of small cracks. In the first case, thirty-two small cracks are considered as shown in Figure 5.13. The second case includes eighty small cracks as depicted in Figure 5.15. Corresponding fracture patterns for these two cases are presented in Figures 5.14 and 5.16, respectively. In both cases, it

can be seen that small cracks behind the macrocrack tip does not effect crack propagation. Furthermore, only small cracks ahead of the macrocrack tip close to the centerline influences the crack propagation pattern and causes crack branching behavior.

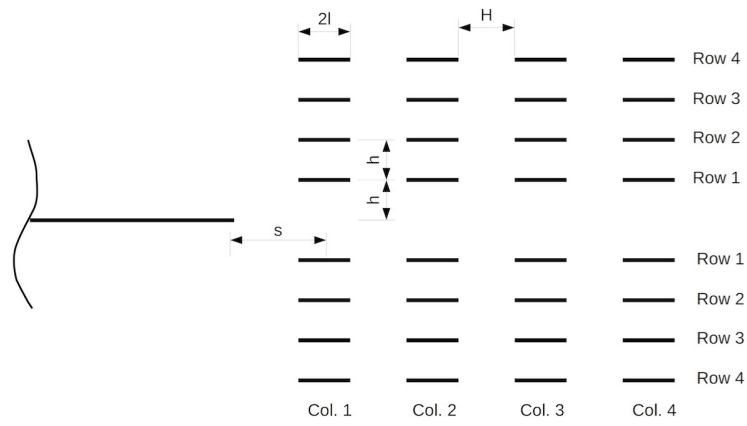


Figure 5.13: Set of vertical small cracks

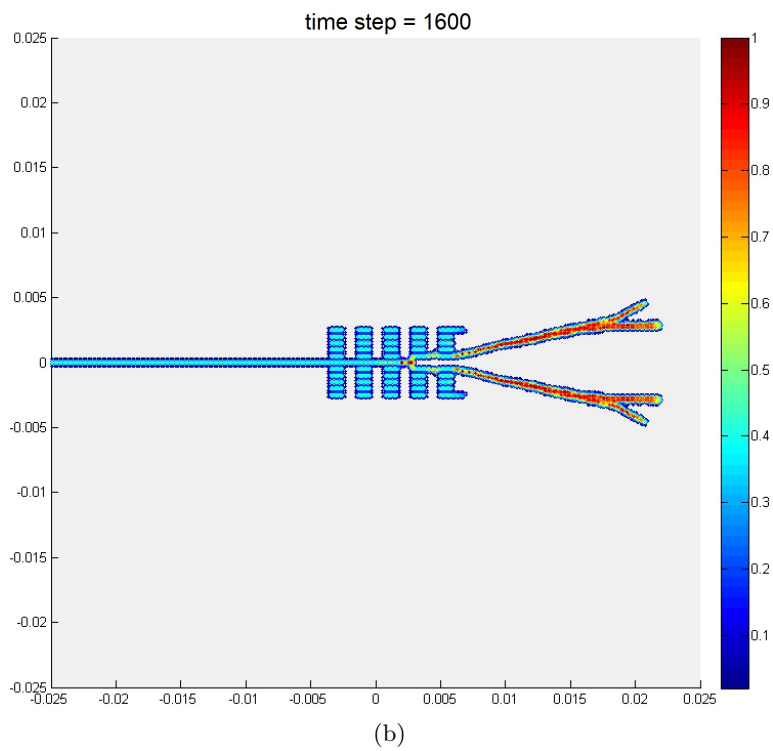
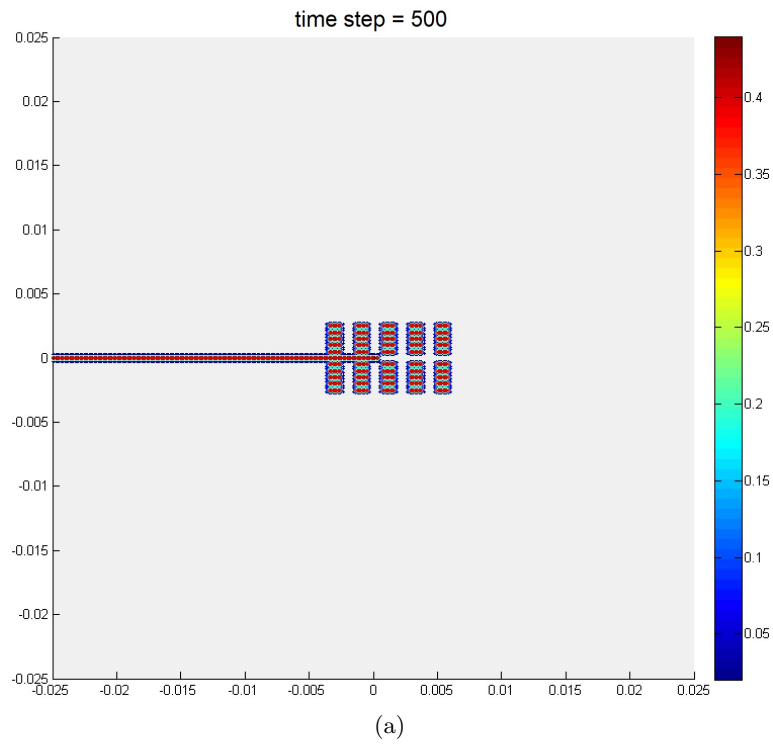


Figure 5.14: (a) Initial configuration and (b) crack propagation in a set with thirty-two small cracks

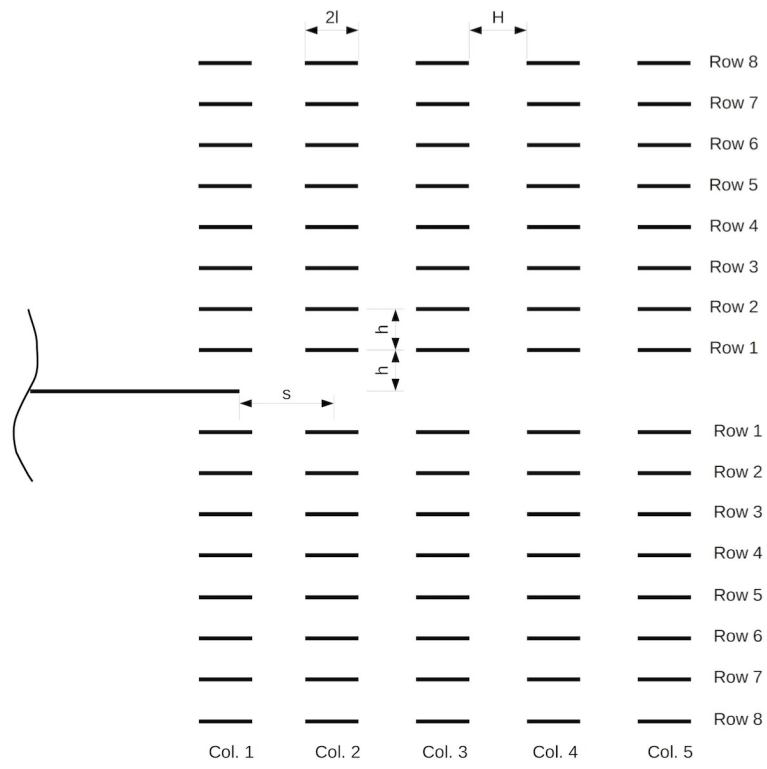
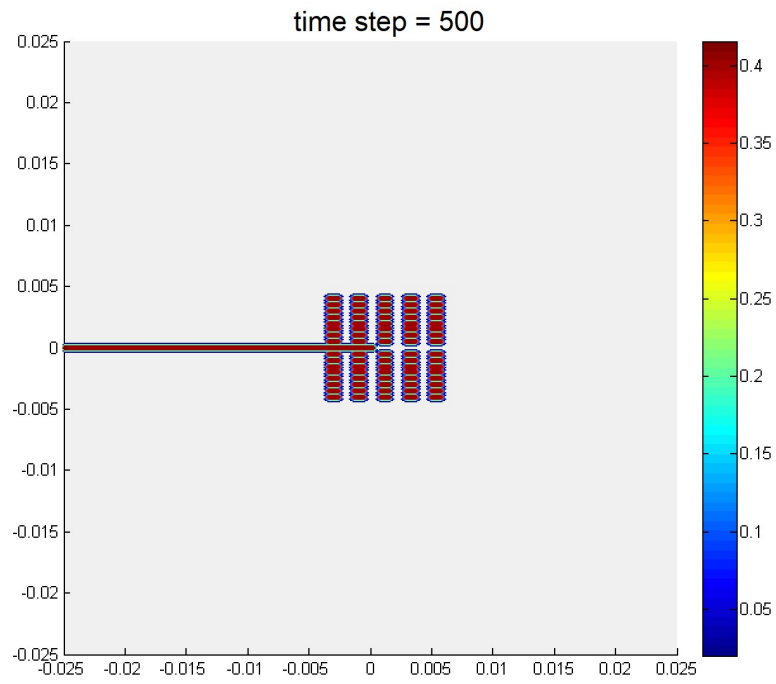
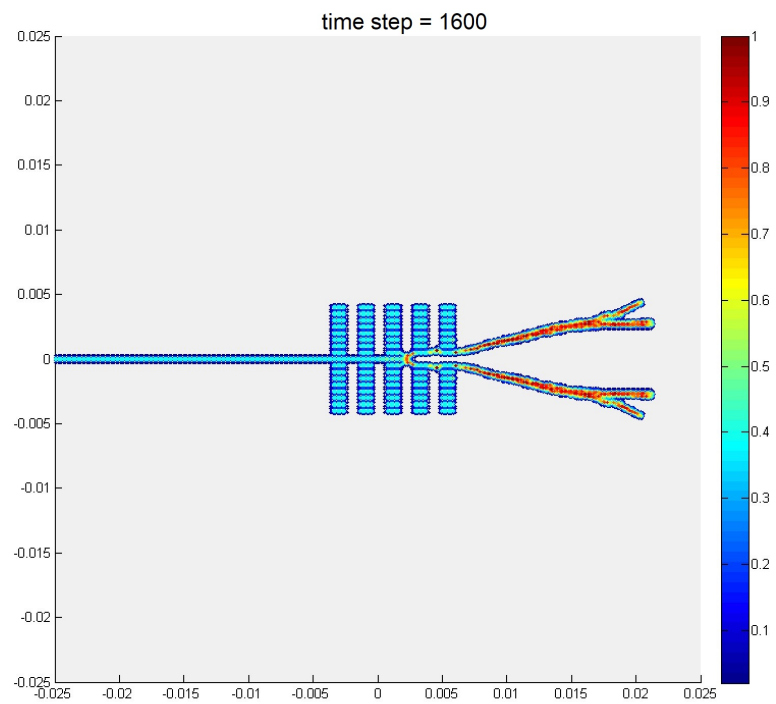


Figure 5.15: Set of eighty small cracks



(a)



(b)

Figure 5.16: (a) Initial configuration and (b) crack propagation in a set with eighty small cracks

5.2.4 Conclusion

In this study, quantitative comparison of effects of small cracks on dynamic macrocrack propagation is obtained by using peridynamics. Several small crack configurations are considered including single small crack collinear to the main crack, two symmetrical small cracks, and horizontal and transverse array of small cracks. The results show expected behaviour when compared to similar cases found in the literature. Which means that certain combinations of small cracks can slow down the crack propagation through the effect of crack shielding and amplification. Finally, two cases including significantly large number of small cracks are demonstrated. These cases show the good capabilities of peridynamics in capturing sophisticated fracture patterns where significantly large number of cracks present. Moreover, presented results also show that this analysis can be useful for the design of new microstructurally toughened materials.

5.3 Peridynamic Model for a Mindlin Plate Resting on a Winkler Elastic Foundation

5.3.1 Introduction

In many engineering applications including marine, civil and transport engineering, analysis of structures resting on an elastic foundation is an important problem of interest (Attar et al. (2014)). To represent the elastic foundation, Winkler and Pasternak formulations are widely utilised. In Winkler formulation, the elastic foundation is represented by distribution of springs to resist the lateral deflection of the structure resting on the elastic foundation. On the other hand, Pasternak formulation can capture the shear interaction between springs (Attar et al. (2014)).

Although there are numerous studies in the literature considering elastic foundation problem, only few of them investigated the behaviour of an existing crack inside a structure resting on an elastic foundation. Amongst these, Matysiak and Pauk (2003) performed stress analysis near a crack tip in an elastic layer resting on a Winkler foundation by using the method of Fourier transforms and dual integral equations.

Farjoo et al. (2012) investigated rolling contact fatigue cracks in railway tracks and used a simplified finite element model (FEM) and extended finite element method (XFEM). They observed that elastic foundation leads to an additional bending stress which increases the crack growth rate significantly. In another study, Attar et al. (2014) investigated the free vibration of a shear deformable beam with multiple open edge cracks using lattice spring model. Finally, Nobili et al. (2014) presented a full-field solution for the linear elasto-static problem of a homogeneous infinite Kirchhoff plate with a semi-infinite rectilinear crack resting on a two-parameter elastic foundation. They calculated stress intensity factors for both symmetric and skew-symmetric loading conditions.

In this study, an alternative approach, peridynamics (Silling (2000)), is used for the analysis of a Mindlin plate resting on a Winkler type elastic foundation. Peridynamics was originally introduced to overcome the limitations of classical continuum mechanics. The equations of motion in peridynamics are in the form of integro-differential equations and they do not contain any spatial derivatives. Therefore, these equations are valid regardless of discontinuities. Peridynamics has been successfully used to analyse different material systems and geometrical configurations (Madenci and Oterkus (2016), Madenci and Oterkus (2017), Oterkus and Madenci (2012), Oterkus et al. (2010), Oterkus et al. (2017), Oterkus and Madenci (2017), Ha and Bobaru (2010), Foster et al. (2010) and Dipasquale et al. (2014)). An extensive literature survey on peridynamics is given in Madenci and Oterkus (2014) and Javili et al. (2019). Aforementioned benefits of peridynamics have attracted interest in solving solid mechanics problems particularly those involving damage and fracture. Majority of such attempts deal with full 3D models or 2D plane stress/plane strain models. There are relatively few peridynamic models considering structures resisting transverse deformation with one dimension (e.g. the thickness) significantly smaller than the other two (e.g. aircraft fuselage, ship hull, pressure vessel etc.) including Silling and Bobaru (2005) for 2D membranes, Taylor and Steigmann (2015), O’Grady and Foster (2014), Diyaroglu et al. (2015) and Reddy et al. (2013) for plates and flat shells, and Chowdhury et al. (2016) for shells. This study is an extension of Mindlin plate formulation developed by

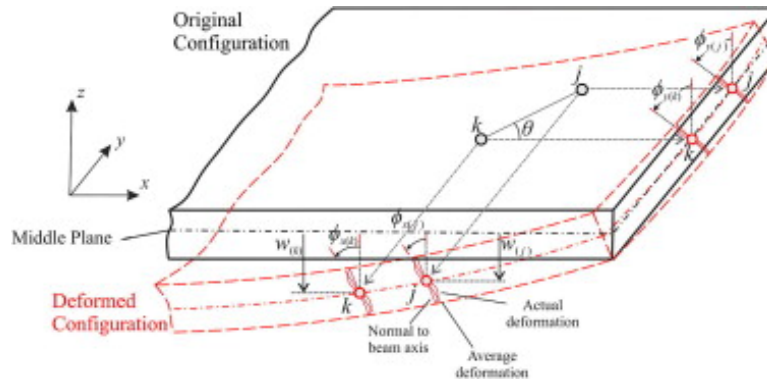


Figure 5.17: Initial and deformed configuration of a Mindlin plate (Diyaroglu et al. (2015))

Diyaroglu et al. (2015). A similar approach was presented by Di Paola et al. (2009) for nonlocal modeling of a beam on an elastic foundation. The current formulation is capable of analysing Mindlin plates resting on an elastic Winkler foundation with damage prediction capability. Moreover, the direct solution approach (Bobaru et al. (2009), Breitenfeld et al. (2014)) is presented to obtain the solution in static conditions rather than using widely adapted Adaptive Dynamic Relaxation (ADR) scheme (Kilic and Madenci (2010)). Finally, several verification and demonstration cases including a Mindlin plate with or without an initial crack subjected to transverse loading or pure bending loading conditions are presented to validate the current formulation and demonstrate its capabilities.

5.3.2 Peridynamic Mindlin Plate Formulation

Peridynamic formulation presented in the previous section is for material points having translational degrees of freedom only. If rotational degrees of freedom are desired to be included to represent Mindlin plate formulation in peridynamics, appropriate changes to the original PD formulation should be made as explained in Diyaroglu et al. (2015). In Mindlin plate formulation, each material point has three degrees of freedom including transverse deflection, w and rotation of planes around x-axis, ϕ_y and y-axis, ϕ_x (see Figure 5.17).

As presented in Diyaroglu et al. (2015), the transverse shear angle and curvature

can be respectively expressed in peridynamic form as:

$$\varphi_{kj} = \frac{w_j - w_k}{\xi_{jk}} - \frac{(\phi_{x(j)} \cos\theta + \phi_{y(j)} \sin\theta) + (\phi_{x(k)} \cos\theta + \phi_{y(k)} \sin\theta)}{2} \quad (5.3)$$

and

$$\kappa_{kj} = \left(\frac{\phi_{x(j)} - \phi_{x(k)}}{|\xi_{jk}|} \right) \cos\theta + \left(\frac{\phi_{y(j)} - \phi_{y(k)}}{|\xi_{jk}|} \right) \sin\theta \quad (5.4)$$

where θ is the peridynamic bond orientation with respect to x -axis. Moreover, the peridynamic equations of motion for the material point k can be derived using the principle of virtual work as:

$$\rho h \ddot{w}_k = c_s \sum_{j=1}^N \phi_{kj} V_j + \widehat{b}_{x(k)} \quad (5.5)$$

$$\rho \frac{h^3}{12} \ddot{\phi}_{x(k)} = c_b \sum_{j=1}^N \kappa_{kj} \cos\theta V_j + \frac{1}{2} c_s \sum_{j=1}^N |\xi_{jk}| \phi_{kj} \cos\theta V_j + \widetilde{b}_{x(k)} \quad (5.6)$$

$$\rho \frac{h^3}{12} \ddot{\phi}_{y(k)} = c_b \sum_{j=1}^N \kappa_{kj} \sin\theta V_j + \frac{1}{2} c_s \sum_{j=1}^N |\xi_{jk}| \phi_{kj} \sin\theta V_j + \widetilde{b}_{y(k)} \quad (5.7)$$

Using transverse shear angle and curvature equations given in Equation 5.3 and Equation 5.4, Equations 5.5 - 5.7 can be rewritten as

$$\rho h \ddot{w}_k = c_s \sum_{j=1}^N \left(\frac{w_j - w_k}{\xi_{jk}} - \frac{(\phi_{x(j)} + \phi_{x(k)})}{|\xi_{jk}|} \cos\theta - \frac{(\phi_{y(j)} - \phi_{y(k)})}{|\xi_{jk}|} \sin\theta \right) V_j + \widehat{b}_{x(k)} \quad (5.8)$$

$$\begin{aligned} \rho \frac{h^3}{12} \ddot{\phi}_{x(k)} = c_b \sum_{j=1}^N & \left[\left(\frac{\phi_{x(j)} - \phi_{x(k)}}{|\xi_{jk}|} \right) \cos\theta + \left(\frac{\phi_{y(j)} - \phi_{y(k)}}{|\xi_{jk}|} \right) \sin\theta \right] \cos\theta V_j + \\ & \frac{1}{2} c_s \sum_{j=1}^N |\xi_{jk}| \left[\frac{w_j - w_k}{\xi_{jk}} - \frac{(\phi_{x(j)} + \phi_{x(k)})}{|\xi_{jk}|} \cos\theta - \right. \\ & \left. \frac{(\phi_{y(j)} - \phi_{y(k)})}{|\xi_{jk}|} \sin\theta \right] \cos\theta V_j + \widetilde{b}_{x(k)} \end{aligned} \quad (5.9)$$

$$\begin{aligned}
\rho \frac{h^3}{12} \ddot{\phi}_{y(k)} = c_b \sum_{j=1}^N \left[\left(\frac{\phi_{x(j)} - \phi_{x(k)}}{|\xi_{jk}|} \right) \cos\theta + \left(\frac{\phi_{y(j)} - \phi_{y(k)}}{|\xi_{jk}|} \right) \sin\theta \right] \sin\theta V_j + \\
\frac{1}{2} c_s \sum_{j=1}^N |\xi_{jk}| \left[\frac{w_j - w_k}{\xi_{jk}} - \frac{(\phi_{x(j)} + \phi_{x(k)})}{|\xi_{jk}|} \cos\theta - \right. \\
\left. \frac{(\phi_{y(j)} - \phi_{y(k)})}{|\xi_{jk}|} \sin\theta \right] \sin\theta V_j + \tilde{b}_{y(k)}
\end{aligned} \tag{5.10}$$

where

$$c_s = \frac{9E}{4\pi\delta^3} k_s^2 \tag{5.11}$$

$$c_b = \frac{E}{\pi\delta} \left(\frac{3h^2}{4\delta^2} + \frac{27}{80} k_s^2 \right) \tag{5.12}$$

and k_s represents the shear correction factor. To describe mode-I and mode-III types of fracture modes, Diyaroglu et al. (2015) defined critical curvature and critical shear angle parameters, respectively, as:

$$\kappa_c = \sqrt{\frac{4G_{Ic}}{c_b h \delta^4}} \tag{5.13}$$

$$\phi_c = \sqrt{\frac{4G_{IIIc}}{c_s h \delta^4}} \tag{5.14}$$

where G_{Ic} and G_{IIIc} represent mode-I and mode-III critical energy release rates, respectively.

5.3.3 Direct Solution of the Peridynamic Mindlin Plate Formulation

In peridynamics, static solution can be obtained by using Adaptive Dynamic Relaxation (ADR) (Kilic and Madenci (2010)) or Direct approach (Bobaru et al. (2009)). In ADR, an artificial damping is introduced to the system and the solution converges to static solution after certain number of iterations. In the direct approach, the inertia term is specified to zero and a matrix solution is required. Therefore, the PD force function can be expressed in terms of the second-order micromodulus tensor \mathbf{C} as (Silling (2000))

$$\mathbf{f} = \mathbf{C}(\xi)\eta \tag{5.15}$$

where

$$\mathbf{C}(\xi) = \frac{\partial \mathbf{f}}{\partial \eta}(0, \xi) \quad (5.16)$$

In the case of PD Mindlin plate formulation, micromodulus tensor, \mathbf{C} can be defined as Jacobian matrix which is a matrix of all first-order partial derivatives of a vector-valued function. Therefore, for the force vector function \mathbf{f} which is a function of shear angle φ and curvature κ , the micromodulus tensor can be defined as:

$$\mathbf{C} = \begin{bmatrix} \frac{\partial f_z}{\partial \varphi} & \frac{\partial f_z}{\partial \kappa} \\ \frac{\partial m_{\phi_x}}{\partial \varphi} & \frac{\partial m_{\phi_x}}{\partial \kappa} \\ \frac{\partial m_{\phi_y}}{\partial \varphi} & \frac{\partial m_{\phi_y}}{\partial \kappa} \end{bmatrix} \quad (5.17)$$

where f_z , m_{ϕ_x} and m_{ϕ_y} represent force or moment functions between material points arising from transverse shear deformation and bending. Utilizing peridynamic equations given in Equations 5.5 - 5.7, force and moment functions can be obtained as:

$$f_z = c_s \varphi \quad (5.18)$$

$$m_{\phi_x} = c_b \kappa \cos \theta + \frac{c_s}{2} |\xi| \varphi \cos \theta \quad (5.19)$$

$$m_{\phi_y} = c_b \kappa \sin \theta + \frac{c_s}{2} |\xi| \varphi \sin \theta \quad (5.20)$$

Therefore, using Equation 5.17 micromodulus tensor \mathbf{C} takes the form of

$$\mathbf{C} = \begin{bmatrix} c_s & 0 \\ \frac{c_s}{2} |\xi| \varphi \cos \theta & c_b \cos \theta \\ \frac{c_s}{2} |\xi| \varphi \sin \theta & c_b \sin \theta \end{bmatrix} \quad (5.21)$$

Substituting Equation 5.21 into Equation 5.15 results in

$$\begin{Bmatrix} f_z \\ m_{\phi_x} \\ m_{\phi_y} \end{Bmatrix} = \begin{bmatrix} c_s & 0 \\ \frac{c_s}{2} |\xi| \varphi \cos \theta & c_b \cos \theta \\ \frac{c_s}{2} |\xi| \varphi \sin \theta & c_b \sin \theta \end{bmatrix} \begin{Bmatrix} \varphi \\ \kappa \end{Bmatrix} \quad (5.22)$$

The force and moment functions between material points j and k can be rewritten by substituting Equation 5.3 and Equation 5.4 into Equation 5.22 as:

$$\begin{Bmatrix} f_z(kj) \\ m_{\phi_x(kj)} \\ m_{\phi_y(kj)} \end{Bmatrix} = \begin{bmatrix} c_s & 0 \\ \frac{c_s}{2} |\xi_{jk}| \varphi \cos\theta & c_b \cos\theta \\ \frac{c_s}{2} |\xi_{jk}| \varphi \sin\theta & c_b \sin\theta \end{bmatrix} \begin{Bmatrix} \frac{w_j - w_k}{\xi_{jk}} - \frac{(\phi_{x(j)} \cos\theta + \phi_{y(j)} \sin\theta) + (\phi_{x(k)} \cos\theta + \phi_{y(k)} \sin\theta)}{2} \\ \left(\frac{\phi_{x(j)} - \phi_{x(k)}}{|\xi_{jk}|} \right) \cos\theta + \left(\frac{\phi_{y(j)} - \phi_{y(k)}}{|\xi_{jk}|} \right) \sin\theta \end{Bmatrix} \quad (5.23)$$

After reorganising Equation 5.23, the following matrix expression of force and moment functions can be obtained as:

$$\begin{Bmatrix} f_z(kj) \\ m_{\phi_x(kj)} \\ m_{\phi_y(kj)} \end{Bmatrix} = \begin{bmatrix} \frac{c_s}{|\xi_{jk}|} & \frac{c_s}{2} \cos\theta & \frac{c_s}{2} \sin\theta \\ \frac{c_s}{2} \cos\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos^2\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta \\ \frac{c_s}{2} \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \sin^2\theta \\ -\frac{c_s}{|\xi_{jk}|} & \frac{c_s}{2} \cos\theta & \frac{c_s}{2} \sin\theta \\ -\frac{c_s}{2} \cos\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos^2\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta \\ -\frac{c_s}{2} \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \sin^2\theta \end{bmatrix} \begin{Bmatrix} w_k \\ \phi_{x(k)} \\ \phi_{y(k)} \\ w_j \\ \phi_{x(j)} \\ \phi_{y(j)} \end{Bmatrix} \quad (5.24)$$

For static and quasi - static problems, the acceleration terms \ddot{w} , $\ddot{\phi}_x$ and $\ddot{\phi}_y$ can be omitted from the equation of motion as:

$$\sum_{j=1}^N \mathbf{f}_{kj} V_j + \mathbf{b}_k = \mathbf{0} \quad (5.25)$$

where $\mathbf{f}_{kj} = [f_{z(kj)} m_{\phi_x(kj)} m_{\phi_y(kj)}]^T$ and $\mathbf{b}_k = [\widehat{b}_k \widetilde{b}_{x(k)} \widetilde{b}_{y(k)}]^T$. Substituting force/moment functions given in Equation 5.24 into Equation 5.25 leads to

$$\sum_{j=1}^N \begin{bmatrix} \frac{c_s}{|\xi_{jk}|} & \frac{c_s}{2} \cos\theta & \frac{c_s}{2} \sin\theta \\ \frac{c_s}{2} \cos\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos^2\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta \\ \frac{c_s}{2} \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| + \frac{c_b}{|\xi_{jk}|} \right) \sin^2\theta \\ -\frac{c_s}{|\xi_{jk}|} & \frac{c_s}{2} \cos\theta & \frac{c_s}{2} \sin\theta \\ -\frac{c_s}{2} \cos\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos^2\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta \\ -\frac{c_s}{2} \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \cos\theta \sin\theta & \left(\frac{c_s}{4} |\xi_{jk}| - \frac{c_b}{|\xi_{jk}|} \right) \sin^2\theta \end{bmatrix} \quad (5.26)$$

$$\left. \begin{array}{l} w_k \\ \phi_{x(k)} \\ \phi_{y(k)} \\ w_j \\ \phi_{x(j)} \\ \phi_{y(j)} \end{array} \right\} V_j + \mathbf{b}_k = \mathbf{0}$$

5.3.4 Peridynamic Mindlin Plate Resting on An Elastic Foundation

In this study, Winkler foundation is considered as the elastic foundation and coupled with PD Mindlin formulation presented in Section 5.3.3. Winkler foundation was originally introduced by Winkler for modelling the soil-structure interactions. Winkler method assumes that vertical translation of the soil, w , at a point depends only upon the contact pressure, p , acting at that point in the idealized elastic foundation and a

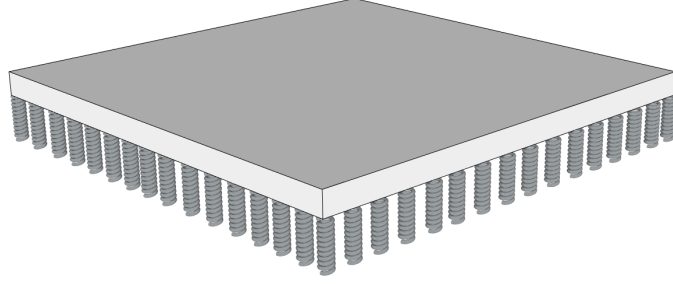


Figure 5.18: Mindlin plate on a Winkler foundation

proportionality constant, k as

$$p = kw \quad (5.27)$$

The proportionality constant, k , is commonly referred to as the modulus of subgrade reaction or the coefficient of subgrade reaction. This model was first used to analyse the deflections and resultant stresses in railroad tracks. In the following years, it has been applied to many different soil/fluid-structure interaction problems, and it is known as die Winkler model.

In order to combine Winkler foundation with PD Mindlin plate matrix formulation, Winkler foundation formulation can be written in matrix form as:

$$\begin{bmatrix} \frac{kh}{V_j} & 0 & 0 & -\frac{kh}{V_j} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} w_k \\ \phi_{x(k)} \\ \phi_{y(k)} \\ w_j \\ \phi_{x(j)} \\ \phi_{y(j)} \end{Bmatrix} + \mathbf{b}_k = \mathbf{0} \quad (5.28)$$

where k is the spring stiffness and h is thickness of the plate. It is assumed that Winkler foundation only affects transverse deflection.

5.3.5 Numerical Results

As part of the numerical results, simple static loading conditions are considered first to compare the PD predictions with the Finite Element Analysis (FEA) results using ANSYS, a commercial FEA software. Next, a plate with a central crack under pure bending resting on a Winkler foundation with very small spring stiffness is considered as a validation case to compare against results obtained in Diyaroglu et al. (2015). Then, fracture behaviour of a pre-cracked ice sheet floating on water under transverse loading condition is investigated.

5.3.5.1 Mindlin Plate Rested on a Winkler Foundation Subjected to Transverse Loading

In the first example, a Mindlin plate rested on a Winkler foundation under half circular edge pressure is considered (see Figure 5.19). This problem was first introduced by Lu et al. (2015b) to simulate displacement distribution for a finite size ice floe interacting with sloping structures.

As it was stated by Lu et al. (2015b), there is no analytical closed-form solution to calculate the deflection and stress distribution of a finite plate with free edges under evenly distributed edge pressure within a half circular area. Therefore, a numerical solution is adopted in order to verify PD results. The length of the square plate is $L = 0.43$ m with a thickness of $h = 0.01$ m. The radius of the loading area is $R = 0.2L$. The Young's modulus of the plate is specified as $E = 5.5$ GPa. Only a single row of material points (collocation) points in the thickness direction is necessary to discretize the domain. The distance between material points is $\Delta x = 0.00215$ m. The loading is applied to a single row of material points at the half circular area as a resultant body load of $\hat{b} = 86.12$ N/m² for the transverse loading. The Winkler foundation modulus to represent the fluid base is $k = \rho_w g = 1025(\text{kg/m}^3) \cdot 9.81(\text{m/s}^2) = 10055.25$ Pa/m, with ρ_w and g being the fluid density and gravitational acceleration, respectively.

The peridynamic solutions for transverse displacement and rotations are compared with finite element solutions obtained by using ANSYS shell element, which is suitable for thick/thin shell structures. As depicted in Figures 5.20-5.21 and Figures 5.22-5.23,

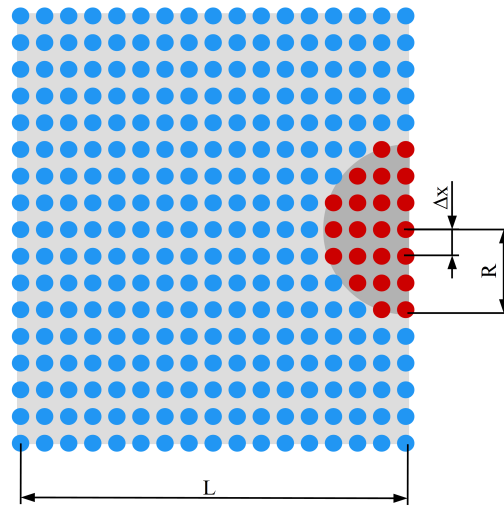


Figure 5.19: Peridynamic discretization of Mindlin plate subjected to the transverse loading (shaded area)

PD and FE solutions are in good agreement with each other and this verifies that the PD direct solution correctly captures the deformation behaviour of the Mindlin plate rested on an elastic foundation.

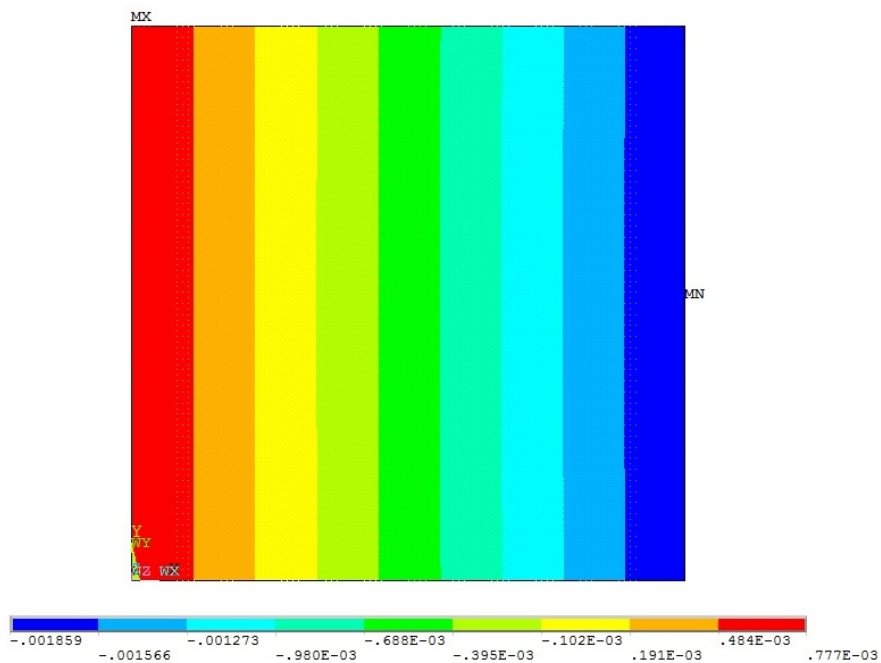


Figure 5.20: FEA results for displacement w

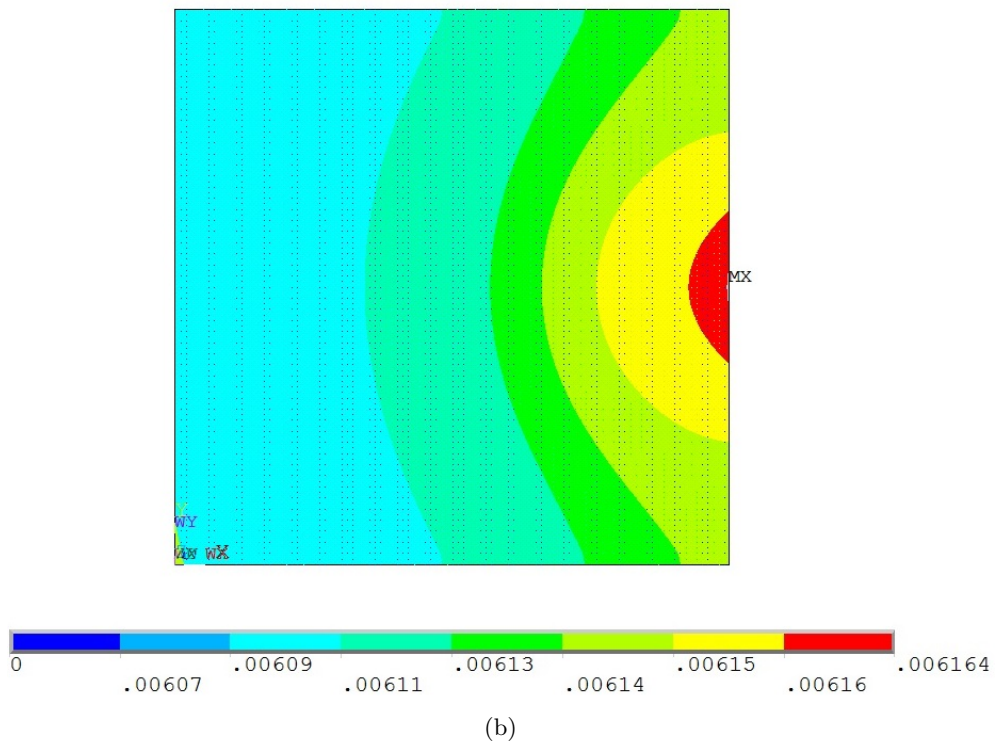
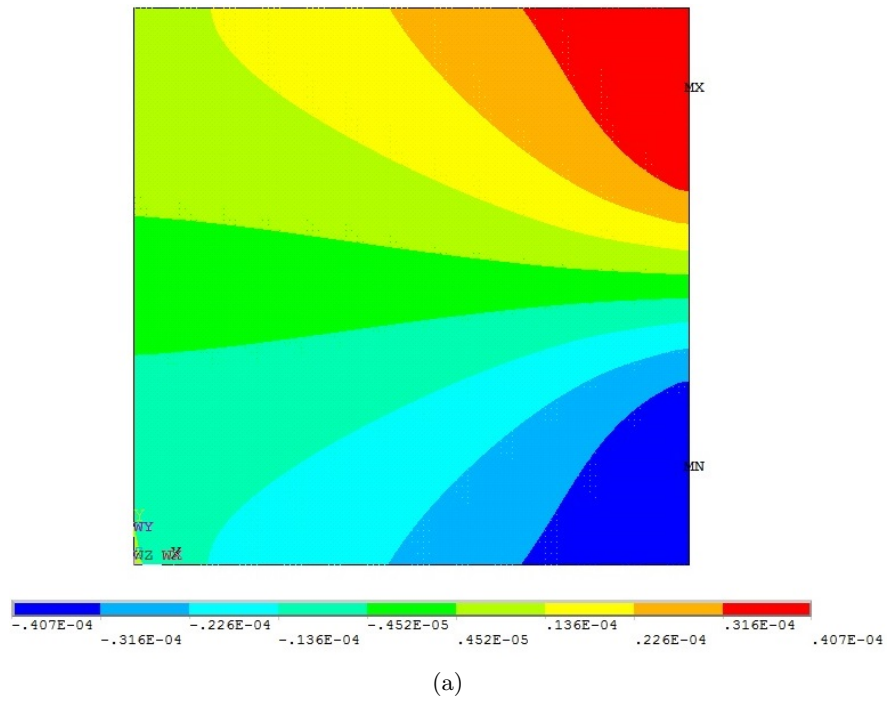
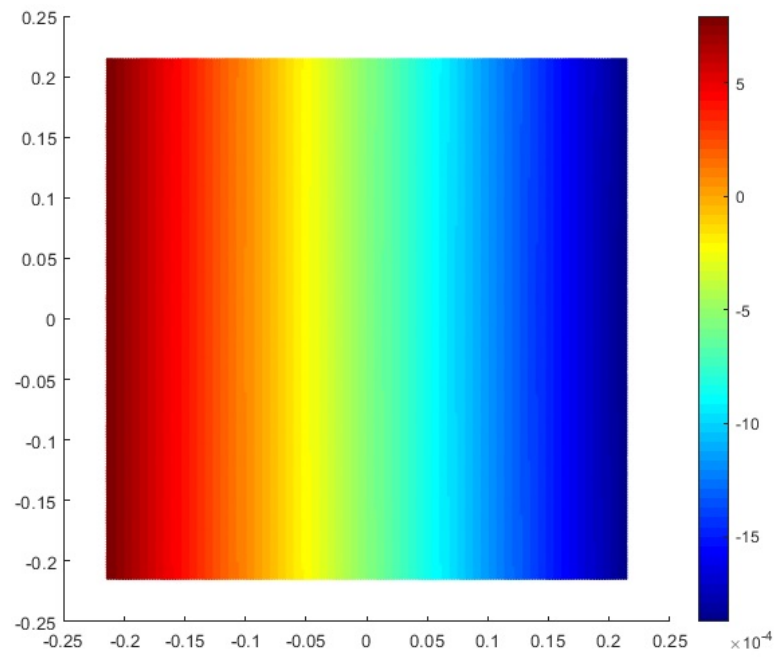
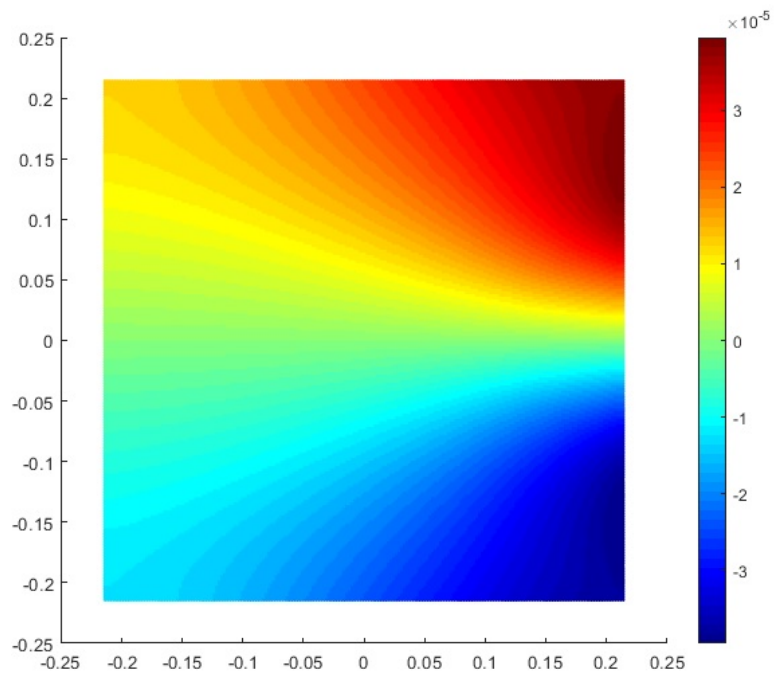
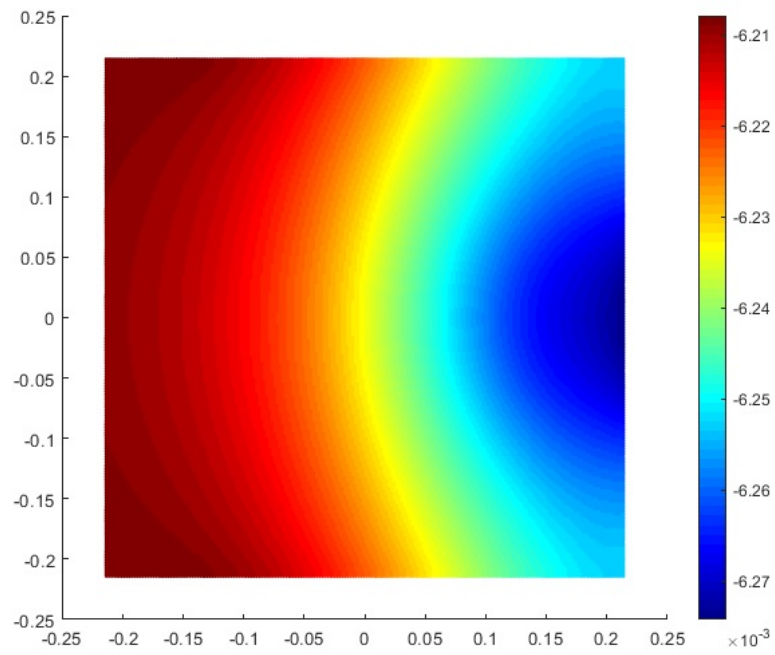


Figure 5.21: FEA results for and rotations (a) ϕ_x and (b) ϕ_y

Figure 5.22: Peridynamic Mindlin plate results for displacement w



(a)



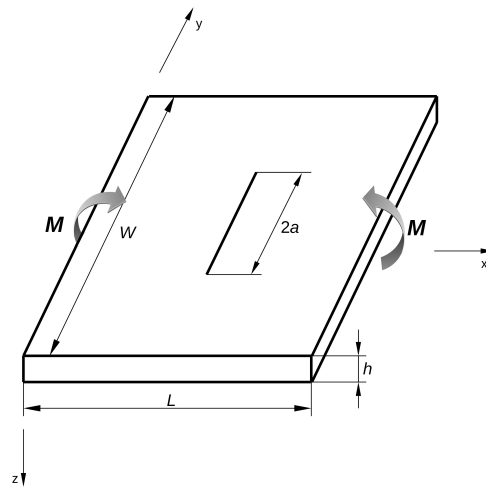
(b)

Figure 5.23: Peridynamic Mindlin plate results for rotations (b) ϕ_x and (c) ϕ_y

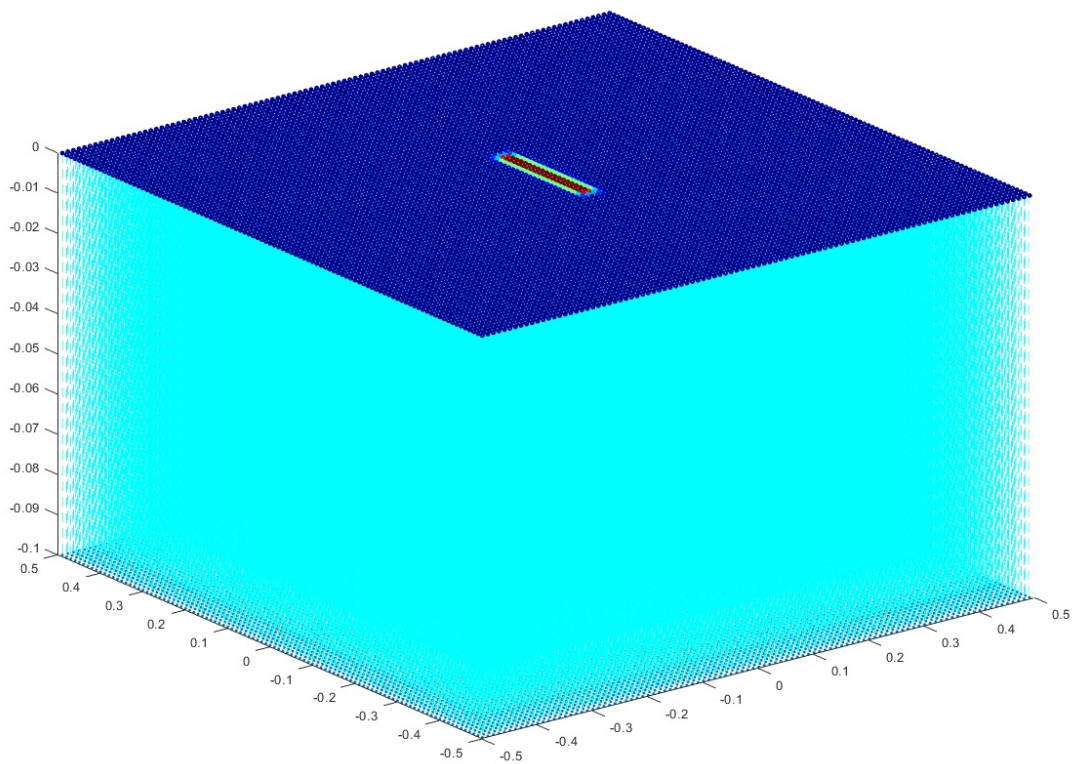
5.3.5.2 Pre-cracked Mindlin Plate Rested on a Winkler Foundation Subjected to Pure Bending Conditions

In the second example, a verification study is considered as in Diyaroglu et al. (2015) to investigate the behaviour of a pre-existing crack in a Mindlin plate. A square plate with an initial central crack aligned with the y-axis is considered as shown in Fig. 8. The length and width of the square plate are $L = W = 1$ m with a thickness of $h = 0.1$ m. Plate thickness to crack length is $h/2a = 0.5$ where $2a$ is the initial crack length.

The Young's modulus of the plate is specified as $E = 3.227$ GPa and shear modulus is $G = 1.21$ GPa. The distance between material points is $\Delta x = 0.01$ m. The horizon size is chosen as $\delta = 3.015 \cdot \Delta x$. The stiffness of the Winkler foundation is set to be very small value, $k = 10^{-9}$ N/m, in order to represent the original example of Diyaroglu et al. (2015) which is free from elastic foundation. The material is chosen as polymethyl-methacrylate (PMMA), which shows brittle fracture behaviour. Mode-I fracture toughness of this material is given as $1.33 \text{ MPa}\sqrt{\text{m}}$ (Ayatollahi and Aliha (2009)). In order to show simple mode-I crack growth, a bending moment loading is applied to a single row of points at horizontal boundary regions of the plate. Small increments of resultant body load of $\Delta\tilde{b}_x = 0.05$ N/m are induced in order to obtain a stable crack growth. Crack starts to grow approximately at $\tilde{b}_x = 284$ N/m as shown in Figure 5.25 and a similar crack pattern is obtained as in Diyaroglu et al. (2015).



(a)



(b)

Figure 5.24: (a) Pre-cracked Mindlin plate under pure bending condition, (b) Peridynamic discretization of pre-cracked Mindlin plate resting on an elastic Winkler foundation

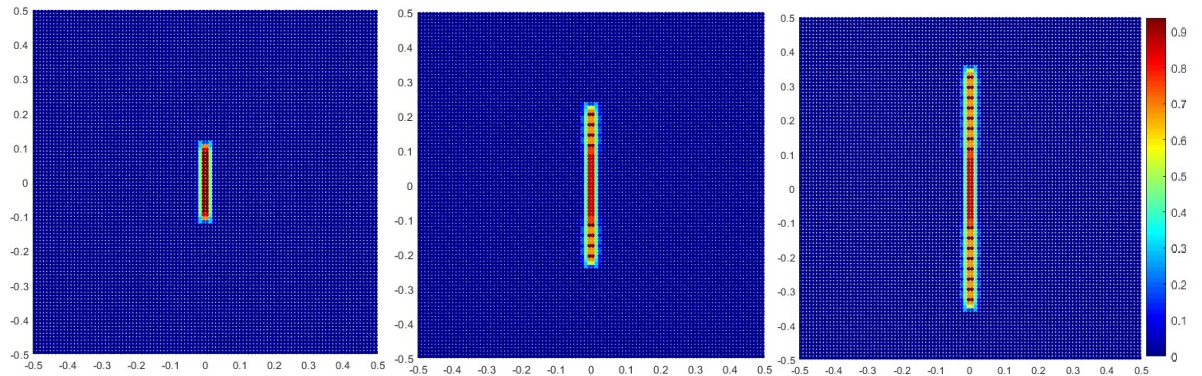


Figure 5.25: Crack propagation for PMMA pre-cracked plate resting on a Winkler foundation

5.3.6 Conclusion

In this study, a new peridynamic model is presented for a Mindlin plate resting on a Winkler type elastic foundation. The formulation is validated by comparing against FEA results for a transverse loading condition for a plate without a crack. For a pure bending loading condition applied to a plate with a central crack free from elastic foundation provided a similar crack pattern that was obtained in an earlier study.

5.4 In-Plane and Out-Of Plane Failure of an Ice Sheet Using Peridynamics

5.4.1 Introduction

Modelling ice-structure interaction is a very difficult process. First of all, many different factors such as strain-rate, temperature, applied-stress, salinity, grain-size, confining pressure and porosity have significant influence on ice material response. Furthermore, macro-scale modeling may not be sufficient to capture the full physical behavior because the microscale effects may have a significant effect on macroscopic material behavior. Hence, it is necessary to utilize a multiscale methodology. To represent the macroscopic ice behavior accurately, Finite Element Method (FEM) has been widely used in the literature. Within FEM framework, various techniques can be used

to model crack propagation including extended finite element method (XFEM) and cohesive zone model (CZM). However, a universally accepted CZM failure model is not currently available and the crack propagation may have mesh dependency. Although, the mesh dependency problem can be overcome by XFEM, enrichment process may lead to an algebraic system with a large number of unknowns which is difficult to solve numerically. Furthermore, since FEM is based on classical continuum mechanics, its formulation do not contain a length scale parameter. Hence, FEM is incapable of capturing phenomenon at the micro-scale. Hence, other techniques should be utilized at the micro-scale and linked to FEM simulation. However, it is not straightforward to obtain a smooth transition between different approaches at different scales. By taking into account all these challenging issues, a new continuum mechanics formulation, peridynamics, can be used for modelling ice failure. Peridynamics is classified as a non-local continuum mechanics formulation and it does not contain spatial derivatives in its formulation. Hence, it is very suitable to predict crack initiation and propagation occurring within the material as the material is subjected to some external loading condition. Furthermore, due to its non-local character, it can capture the phenomenon at multiple scales.

There exists significant number of studies in the literature focusing on in-splitting failure of ice sheets. Moreover, there is a large volume of experimental studies done in the field or multiple lab tests. Timco (1987) has done a series of indentation and penetration tests on a floating sheet of ice. During these tests, splitting of an ice sheet was identified and named as “radial cracks”. Although this was an extensive study, it still lacked information on the size and geometry of the ice sheet since most of the effort was focused on indentation rate and aspect ratio. Similarly, Grape et al. (1992) investigated the influence of lateral confinement on failure patterns and indentation pressure. Most of the aforementioned studies when used for splitting problems were obscured by many ambiguities, until Dempsey et al. (1999) conducted a series of in-situ fracture tests on edge-cracked square plates (sized from 0.5-80 m) together with nonlinear fracture mechanics (NLFM) analysis. This study provides a clear picture on splitting loads and the scale effect on fracture toughness and ice strength. Another

paper that is very important is by Lu et al. (2015a) where the authors conducted an extensive research on splitting failure of an ice sheet together with conducting an exhaustive overview of different methods used to calculate the splitting loads, such as, LEFM, Cohesive Zone, and Plastic limit theory.

On the other hand, out-of-plane failure of an ice sheet has also been studied extensively in the literature either experimentally or theoretically including Ashton (1986), Kerr (1976), Langhorne et al. (1999), Michel (1978a), Sodhi (1995) and Squire et al. (2012). The focus of this research has been mainly related to the interaction of sloping structures and ice in ice infested waters. In arctic marine environment offshore structures and ships have specific design. For offshore structures those are usually sloped pylons and for ships/icebreakers specific shape of the bow. The reason behind sloping geometry is to introduce a vertical load on the edge of the incoming ice floe in order to force the ice into a bending failure mode. Bending failure mode induced in such a way can be described as some type of out-of-plane failure mode. For all intents and purposes, numerical ice sheet models are usually represented as an infinitely long thin plate resting on a Winkler elastic foundation.

In this study, two specific examples are investigated; in-plane failure and out-of-plane failure of an ice sheet subjected to an edge load. These two cases can be derived from the same real life example; an ice sheet interacting with a sloping structure. As it was described by Lu et al. (2015b) only initial contact between the ice sheet and the sloping structure is taken into account (see Fig 1). When ship's hull interacts with an ice sheet a complex stress condition will develop which can be represented by four different load components as shown in Figure 5.26. Loads in y direction are ones inducing the in-plane failure of an ice sheet. The load component along z -direction is responsible for the out-of-plane failure of ice sheet. The in-plane compressive stress within the ice sheet increases due to the load component along x -direction. Although ice failure condition and the fracture patterns will be affected by loads in all three directions, interactions between loads are ignored and they are decoupled into an in-plane and out-of-plane problem. Hence, in this study a simplified decoupling approach is utilized by constraining the in-plane failure problem to only F_y load and out-of-plane failure

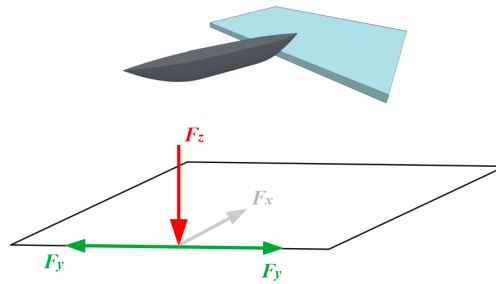


Figure 5.26: Contact and load description

problem to F_z load. For the in-plane failure problem, a 2 dimensional bond based peridynamic model is implemented and for out-of-plane failure problem, 2 dimensional plate is modeled as a Mindlin plate resting on a Winkler elastic foundation. The idea behind this study is to use peridynamic theory to predict ice splitting load for in-plane fracture of finite ice sheet. Moreover, the relationship between radial crack initiation load and measured out-of-plane failure load corresponding to the eventual failure of the ice sheet is established.

5.4.2 2D Bond Based Peridynamic Model for In-Plane Failure of an Ice Sheet

As mentioned earlier, in-plane splitting failure of an ice sheet has been extensively studied in the literature. There is a large volume of experimental studies done in the field or multiple laboratory tests. Amongst these studies, experiments that dealt directly with splitting of the ice sheet were only done by Dempsey et al. (1999). In their in-situ fracture tests, a series of edge-cracked square plates (sized from 0.5-80 m) were torn apart by using a hydraulically driven flatjack. Moreover, they studied the effect of ice sheet size on fracture properties (such as the fracture toughness, traction and separation law for ice, and evolution of the failure process zone) using known histories of load-displacement information and relevant crack opening displacement. The idea behind this analysis stems from Lu et al. (2015a) where the main goal is to obtain the ice splitting load utilizing analytical and numerical methods and compare the results with the experimental results of Dempsey et al. (1999). Difference between Lu et al.

(2015a) and this study is in methods used for comparison with the experimental data. They used several methods, such as CZM + weight function, LEFM + weight function, Plastic limit theory etc., where in this study 2D Bond Based Peridynamic model was applied.

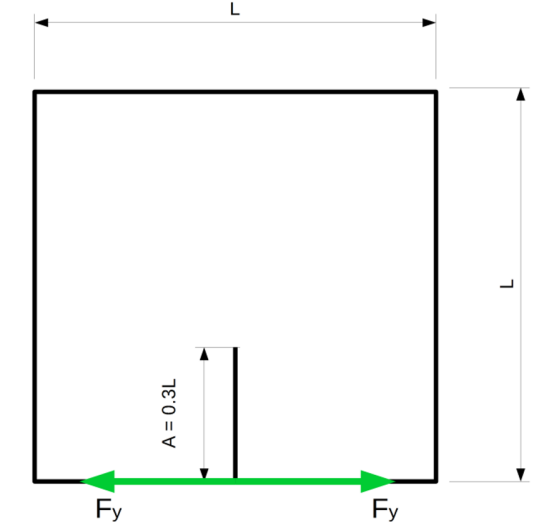


Figure 5.27: Illustration of the model utilized to study in-plane failure of ice plate

Plate geometry and crack size are defined in Figure 5.27. Parameters used to generate peridynamic model for this study is given in Table 5.5, with a fixed horizon size of $\delta = 3\Delta x$, where Δx is the distance between material points. Ice is modelled as an isotropic material with Young's modulus of $E = 5$ GPa with ice thickness of $h = 1.8$ m. For this example sea ice is considered as a brittle material. Mode-I fracture energy of sea ice is given as 15 J/m².

Table 5.5: Test model dimensions and peridynamic inputs

	$L = 3[m]$	$L = 10[m]$	$L = 30[m]$	$L = 80[m]$
$A[m]$	0.3	3	9	24
$dx[m]$	0.015	0.05	0.15	0.4
$dt[s]$	1	1	1	1
Num. PD points	40000	40000	40000	40000
$\sigma_{load}[kPa/s]$	12.965	7.101	4.1	2.511

Loading rate was obtained the same way as it was done by Lu et al. (2015a). In

this study, point load was assumed as the loading with a loading rate equivalent to the loading rate of the flatjack. From Figure 6 in paper by Mulmule and Dempsey (1999), the loading rate of the flatjack can be obtained as 0.41 kPa/s for a size of 3030 m² ice sheet. It is also assumed that as the size of the ice sheet decreases, loading rate increases. In other words, the smallest ice sheet will have the fastest loading rate (in our case 3 m). This leads us to the following relationship:

$$\dot{\sigma}_{test_3} = \dot{\sigma}_{test_{30}} \times \sqrt{30/3} = 1.2965[\text{kPa/s}] \quad (5.29)$$

In our case loading rate was increased by a factor of 10 as it can be seen from Table 5.5. This was done in order to decrease computational time.

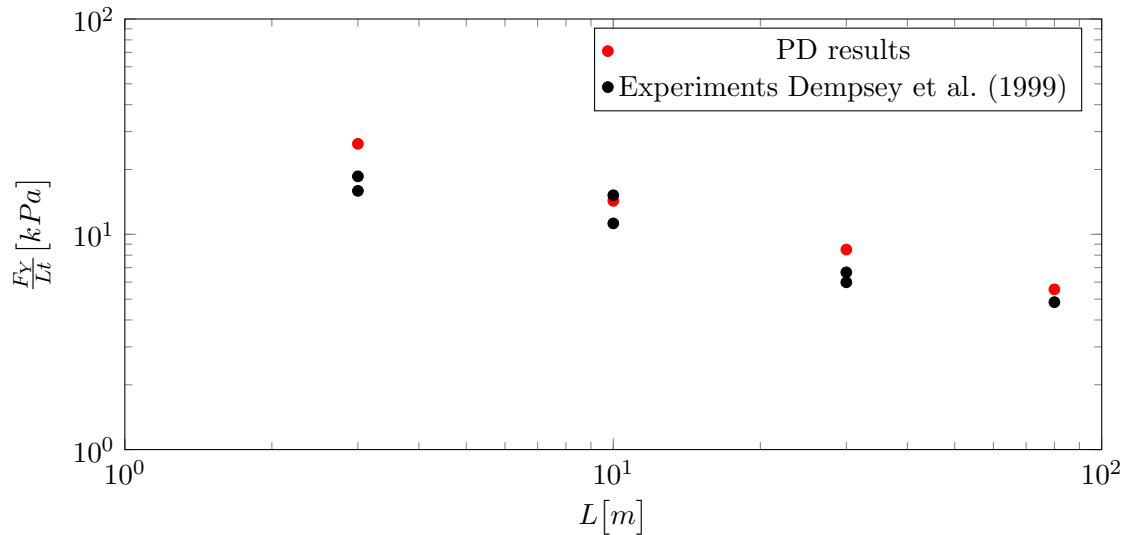


Figure 5.28: Splitting load comparison between PD results and experiments

As it can be seen from the Figure 5.28 results obtained for the splitting load of the ice plate from the 2D Bond Based Model have a relatively good agreement with the experimental data. Furthermore, peridynamic results agree well when compared to LFEM + weight function and CZM + weight function results from Lu et al. (2015a), shown in Figure 14.

5.4.3 Peridynamic Mindlin Plate on Winkler Foundation Model for Out-Of-Plane Failure of an Ice Sheet

This out-of-plane failure case was originally considered by Lu et al. (2015b), where the ice plate interacting with the conical body is simplified by representing the ice sheet with a thin plate resting on an Winkler elastic foundation subjected to an evenly distributed edge pressure inside a half circular area. In their study, Lu et al. (2015b) tried to determine the size of the ice sheet which can be considered either a finite size ice sheet or a semi-infinite ice sheet. The distinction between finite and semi-infinite sheet sizes was classified by following definition:

1. finite size ice floes are broken by radial crack initiation and propagation
2. semi-infinite ice floe is broken by sequentially forming radial and circumferential cracks where following relationship holds (originally from Equation 77 of Kerr (1976)) $F_{Z,B,semi}^{test} \approx 1.6F_{Z,R0,semi}$.
 - $F_{Z,B,semi}^{test}$ is measured breakthrough load corresponding to the eventual out-of-plane bending failure of a semi-infinite ice floe
 - $F_{Z,R0,semi}$ is maximum load required to initiate the radial crack in an ice floe (see Lu et al. (2015b))

Authors were able to establish the difference between size of finite and semi-finite ice sheet by studying only finite size ice sheet which they theoretically formulated and then solved it numerically. What they didn't show were actual fracture patterns for the different size ice sheets and also didn't establish the relationship between maximum load required to initiate the radial crack and measured out-of-plane bending failure load for a semi-infinite ice sheet. Fracture patterns have been studied in a separate paper, see Vazic et al. (2019b). In our work we have tried to capture the relationship described by Kerr (1976) analyzing several different ice sheet sizes.

In Lu et al. (2015b) the ice plate was modelled by using Kirchhoff's plate bending theory and elastic foundation was defined with Winkler foundation model, where the Kirchhoff's plate bending theory was solved with Finite Element Method (FEM). For

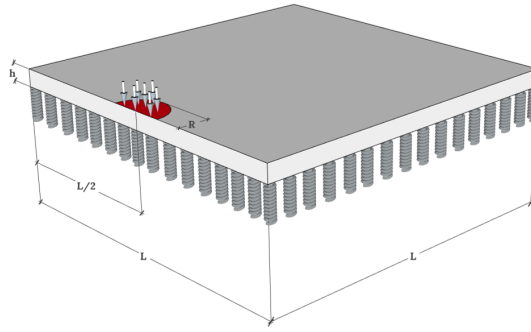


Figure 5.29: Model outline for out-of-plane failure of ice sheet

the finite size ice sheet, Lu et al. (2015b) defined two length scales including the characteristic length l (given in Equation 5.30) and the physical length L , where physical length is defined as $L = n \times l$ with n being a non-dimensional factor. Hence, the characteristic length can be written as:

$$l = \sqrt[4]{\frac{D}{k}} \quad (5.30)$$

where D is the flexural rigidity of the plate defined as

$$D = \frac{Et^3}{12(1-\nu^2)} \quad (5.31)$$

and k is the foundation modulus (defined in Equation 5.32) for the fluid base, with ρ_w and g being the fluid density and gravitational acceleration, respectively.

$$k = \rho_w g \quad (5.32)$$

This study is based on Mindlin plate formulation developed by Diyaroglu et al. (2015) and Mindlin plate resting on Winkler foundation developed by Vazic et al. (2019c). The formulation is capable of analyzing Mindlin plates resting on an elastic Winkler foundation with damage prediction capability. Moreover, the direct solution approach (Bobaru et al. (2009)) is used to obtain the solution in static conditions rather than using widely adapted Adaptive Dynamic Relaxation (ADR) scheme (Kilic and Madenci (2010)).

According to Lu et al. (2015b) and compared to Nevel's solution (Nevel (1965)) normalized radial crack initiation load is approaching solution of a semi-infinite plate when the ice sheet size is $L \geq 4l$. Effectively, this implies that square finite ice sheet can be approximated as a semi-infinite ice sheet when its physical size is 4 times bigger than its characteristic length l . For our analysis this means that plates bigger than $4l$ should mirror the relationship established by Kerr [6] - $F_{Z,B,semi}^{test} \approx 1.6F_{Z,R0,semi}$

Within our example, we have considered several semi-infinite ice sheet lengths. Ice sheet length is defined by non-dimensional factor $n = 5,7,9,12,15$ where $L = n \times l$. Load area radius representing the sloping structure load is set to $R = 0.2l = 0.086$ m. The thickness of the plate is $h = 0.01$ m (Figure 5.29).

Ice is modelled as an isotropic material with Young's modulus of $E = 5.5$ GPa and shear modulus of $G = 2.0625$ GPa. The distance between material points is $\Delta x = 0.01935$ m. The horizon size is chosen as $\delta = 3\Delta x$. Winkler foundation stiffness k is set to $k = 1.0055$ N/m which roughly approximates water behavior. For this example sea ice is considered as a brittle material. Mode-I fracture toughness of sea ice is given as $0.06 \text{ MPa}\sqrt{m}$ (Schulson and Duval (2009)). To the authors knowledge there is no available value for Mode-III fracture toughness of sea ice in the current literature. We assumed Mode-III toughness to be 7 times greater than Mode-I by comparing the ratios with other brittle materials such as PMMA.

In order to establish initial damage so that crack propagation can occur and also satisfy "first crack condition" (so that we can measure radial crack initiation load, $F_{Z,R0,semi}$) we broke 5% of bonds with the highest Von Mises stress. This percentage was chosen because if more than 5% of bonds are broken, unstable fracture is observed and if less than 5% of bonds are broken, crushing behavior is observed, which is followed by unstable fracture. Initial load is set to 0 and then small increments of resultant body load $\Delta \hat{b}_z = 0.1 \text{ N/m}^2$ are induced to obtain a stable crack growth.

It can be clearly seen from the Figure 5.30 that if the ice sheet length increases and by doing so better approximates semi-infinite behavior, the results are approaching the relation established by Kerr (1976). This means that peridynamic Mindlin plate on Winkler foundation model not only captures correct fracture patterns presented in

Vazic et al. (2019b) but also is able show correct relation between measured out-of-plane bending failure load for a semi-infinite ice sheet $F_{Z,B,semi}^{test}$ and maximum load needed to initiate the radial crack in an ice sheet $F_{Z,R0,semi}$

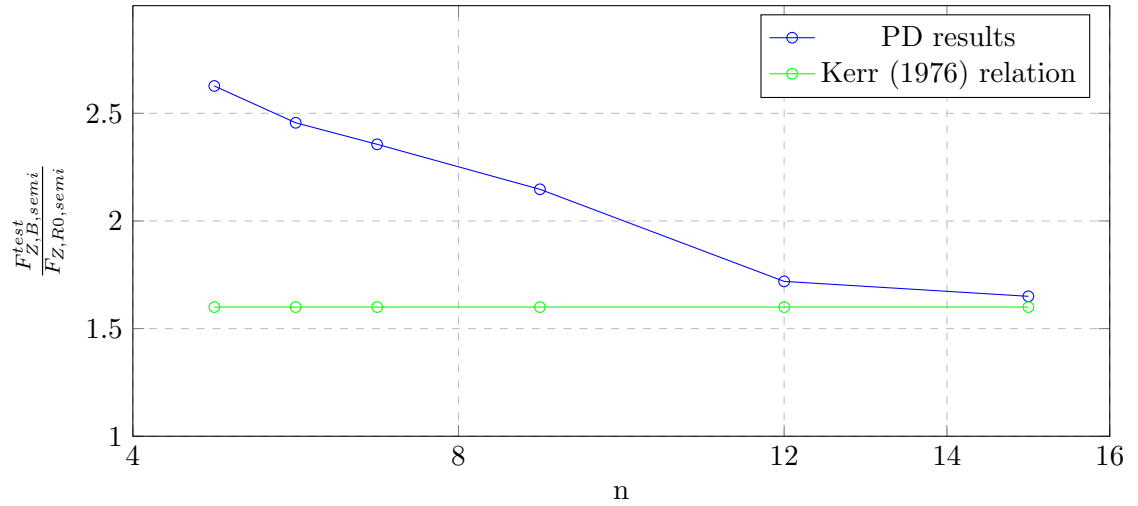


Figure 5.30: Comparison between Kerr's relation and PD results for different size semi-infinite plates

5.4.4 Conclusion

In this study, two different peridynamic models for ice fracture are presented. First model is a 2 dimensional Bond Based model which is used to calculate in-plane splitting loads for several differently sized ice plates, where it was shown that the model has a good agreement with the experimental data. Second model is Mindlin plate resting on a Winkler type elastic foundation used to prove the relation established by Kerr (1976). As it can be seen from the results Mindlin plate model is able to establish such relation if the plate length approaches semi-infinite ice sheet size.

5.5 Peridynamic Approach for Modelling Ice-Structure Interactions

5.5.1 Introduction

Despite of its advantages, utilization of the Arctic region for sailing brings new challenges due to its harsh environment. Therefore, ship structures must be designed to withstand ice loads in case of a collision between a ship and ice takes place. Although experimental studies can give invaluable information about ship-ice interactions, full scale tests are very costly to perform. Therefore, computer simulations can be a good alternative. As it was mentioned in the previous Section 5.4 ice-structure interaction modelling is a very challenging process, where ice material response depends on many different factors and micro-scale effects may have a significant effect on macroscopic material behaviour. Hence, it is necessary to utilize a multi-scale methodology. However, multi-scale approach is not straightforward process as it is difficult to obtain a smooth transition between different scales. With this in mind peridynamics will be utilized for ice fracture modelling as it was done in the previous study.

Out-of-plane failure of an ice floe has been an important research topic for decades. The main focus of this study is related to the interaction of sloping structures and ice floe (e.g., icebreakers, fixed and floating offshore structures) in ice infested waters. When comparing ice floe's failure processes and failure pattern for different problems, such as the "ice-sloping structure interaction problem" and "ice cover's bearing capacity problem", many similarities can be observed (Kerr (1976)). In both cases, a two stage fracture of an ice floe was observed. The first stage is the so-called radial cracking of the ice floe (i.e., radial cracks emanating from the vertically loaded area) and the second stage is the formation of circumferential cracks some distance away from the vertically loaded area.

In this study, peridynamic modeling of an ice floe interacting with sloping structure is presented as a 2D and 3D model. The two stage fracture of an ice floe is investigated by considering a 2D plate modeled as a Mindlin plate on a Winkler foundation and a 3D model based on a bond based peridynamic model with a rudimentary buoyancy

represented as a body load. First model was originally considered by Lu et al. (2015b), where the ice plate interacting with the conical body is simplified by representing the ice floe with a thin plate resting on an elastic foundation under an evenly distributed edge pressure within a half circular area. The idea behind Lu et al. (2015b) paper was to answer the following question: “how small/large should an ice floe be to be treated as a finite size/semi-infinite ice floe”. Second model is more similar to the real life example as the conical rigid body is a relatively good approximation of the ship’s bow, but this model is a purely qualitative example as we are only trying to show fracture patterns that can be found in the literature and to show peridynamic capabilities when dealing with impact problems. There is no analysis of forces exerted by ice onto the rigid body nor are there any analyses of stresses within the ice sheet. The idea behind these studies is development of in-depth ice structure interaction model that can be beneficial for future re-search that will better describe mechanisms related to ice failure and subsequently enable design of the more optimal structures for polar environments.

5.5.2 2D Bond Based Peridynamic Mindlin Plate on Winkler Foundation

This case study is a continuation of the problem from the Section 5.4 where the ice plate interacting with the conical body is simplified by representing the ice floe with a thin plate resting on an elastic foundation under an evenly distributed edge pressure within a half circular area. As it was mentioned in the previous Section, Lu et al. (2015b) tried to answer the following question: “how small/large should an ice floe be to be treated as a finite size/semi-infinite ice floe“. The distinction between finite and semi-infinite floe sizes can be classified as:

- finite size ice floes are broken by radial crack initiation and propagation
- a semi-infinite ice floe is broken by sequentially forming radial and circumferential cracks

Authors were able to establish the difference between size of finite and semi-finite ice floe, by studying only finite size ice flow which they theoretically formulated and

then solved it numerically. What they didn't show were actual fracture patterns for the different size ice floes. In our work we have tried to capture different fracture patterns depending on the floe size in order to distinguish between the finite and semi-infinite ice floes.

As in the previous example, we have considered several finite ice floe lengths. Ice floe length is defined by non-dimensional factor $n = 1,4,5,6,7,9$ where $L = n \times l$. Load area radius representing the sloping structure load is set to $R = 0.2 \cdot l = 0.086$ m. The thickness of the plate is $h = 0.01$ m (Figure 5.32).

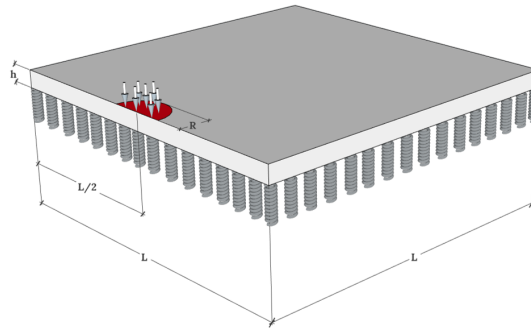


Figure 5.31: Illustration of the model utilised to study ice floe failure

Ice is modelled as an isotropic material with Young's modulus of $E = 5.5$ GPa and shear modulus of $G = 2.0625$ GPa. The distance between material points is $\Delta x = 0.01935$ m. The horizon size is chosen as $\delta = 3.015\Delta x$. Winkler foundation stiffness k is set to $k = 1.0055$ N/m which roughly approximates water behaviour. For this example sea ice is considered as a brittle material. Mode-I fracture toughness of sea ice is given as $0.06 \text{ MPa}\sqrt{\text{m}}$ (Schulson and Duval (2009)). We assumed Mode-III toughness to be 7 times greater than Mode-I by comparing the ratios to other brittle materials such as PMMA.

As in the previous Section 5.4, 5% of bonds with the highest Von Mises stress were broken in order to establish initial damage so that crack propagation can occur. Initial load is set to 0 and then small increments of resultant body load $\Delta \hat{b}_z = 0.1$ N/m are induced in order to obtain a stable crack growth.

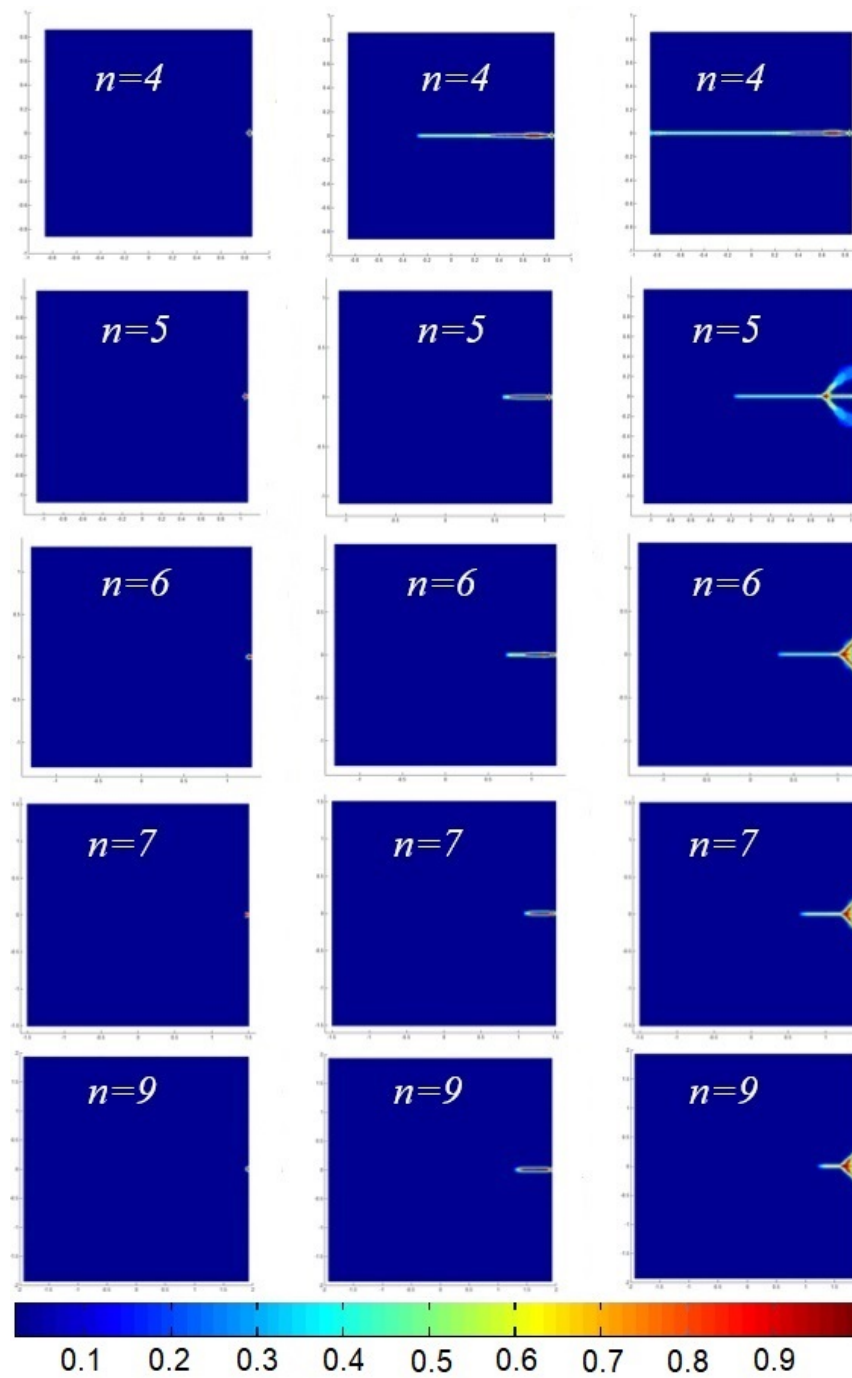
According to Lu et al. (2015b) and compared Nevel's (Nevel (1965)) solution to demonstrate that the normalised radial crack initiation load decreases rapidly towards

the solutions of a semi-infinite plate when the floe size $L \leq 4l$. In practice, this implies that for a square finite ice floe with its physical size 4 times bigger than its characteristic length l , it can be approximated as a semi-infinite ice floe. For our analysis this means that plates bigger than $4 \times l$ should show fracture behaviour identical to those of a semi-infinite plate as it is stated by Kerr (1976): “Observations in the field indicate that the failure mechanism of a semi-infinite plate subjected to a force P at the free edge proceeds as follows: First a radial crack forms, which starts under the load and propagates normal to the free boundary. This is followed by the formation of a circumferential crack that causes final failure.” This is clearly shown in Figure 5.32. Furthermore, for a small size ice floe, in addition to the energy needed to arouse sufficient flexural deformation, another type of deformation (e.g., rigid body rotation) is also present because of the free boundaries. This behaviour can be clearly seen in Figure 5.33, where there is no crack propagation and instead only plate rotation is present.

5.5.3 3D Bond Based Peridynamic Model with Rigid Body Impact Load

The main idea behind this model is to represent idealised real life ice structure interaction where the rigid conical body is interacting with a deformable ice plate. Furthermore, we’ve wanted to test our peridynamic code on very large models in order to see the effects of parallelization on the computational time. The ice plate is based on a 3D bond based peridynamic model for isotropic material. The model is solved by using explicit peridynamic solution. Model parameters for ice, plate and conical body are shown in Table 5.6, Lubbad and Løset (2011). The distance between material points is $\Delta x = 0.11$ m, time step size is $dt = 3.21 \cdot 10^{-4}$ s and number of time steps is $n = 100000$. Buoyancy is represented by a rudimentary body load that is calculated by using Archimedes’ law (calculated at the initial time step) and the plate is constrained in x and y directions at the left and right edges.

As this is a large model with 629265 material points and 47155770 bonds, we had to use a parallel implementation of our peridynamic code which yields a total run time of 11 h 54 m or 0.42 s per time step. Comparing to sequential code, we obtained a

Figure 5.32: Ice failure for $n = 4, 5, 6, 7$, and 9

total time reduction of a factor of 5. Note that there is no initial damage in the model. Expected fracture behaviour is given in the experimental study by Lubbad and Løset

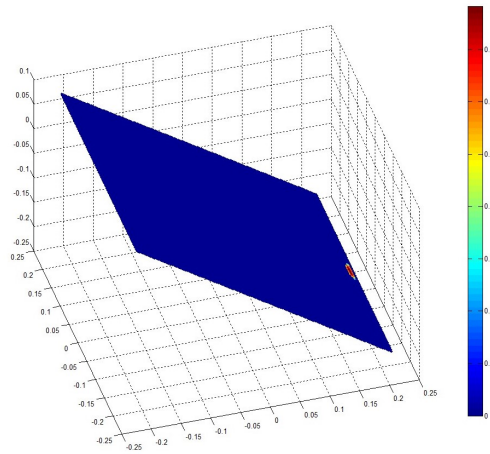


Figure 5.33: Plate rotation for $L = 1 \times l$

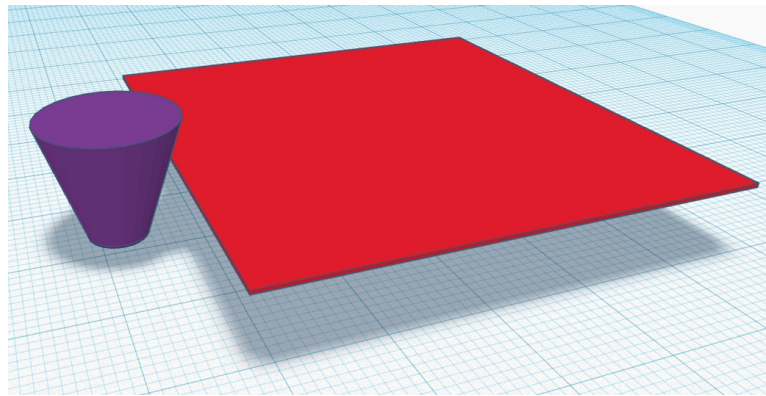


Figure 5.34: 3D Model

(2011) (see Figure 5.35).



Figure 5.35: Ice fracture behaviour

As shown in Figures 5.36 - 5.38, our model is able to capture the behaviour of semi-infinite ice plate, observed by Kerr (1976) and Lubbad and Løset (2011). This

Table 5.6: Problem setup parameters

Problem parameters	
Ice thickness	0.33 m
Ice width	50 m
Cone waterline width	6.8 m
Cone angle	61.4°
Cone height	4.54 m
Cone height above water	1.7 m
Interaction speed	1.202 m/s
Young's modulus of ice	0.35 GPa
Fracture energy	15 J/m ²

behaviour can be described as: first a radial crack starts to propagate which is followed by a circumferential crack that will then lead to complete loss of bearing capacity.

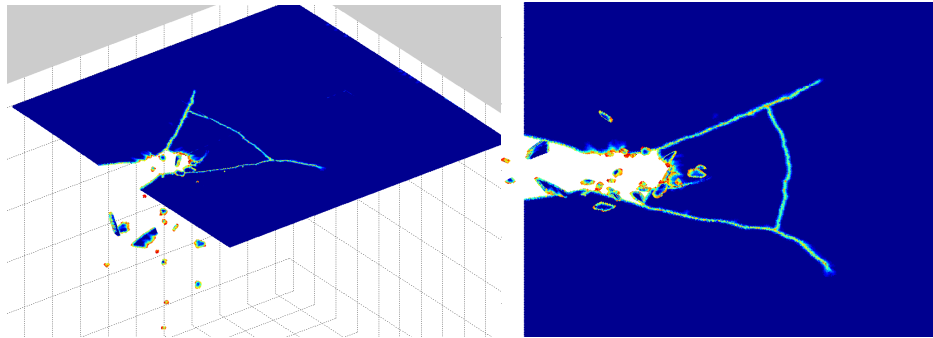


Figure 5.36: Ice fracture at time step of 30000

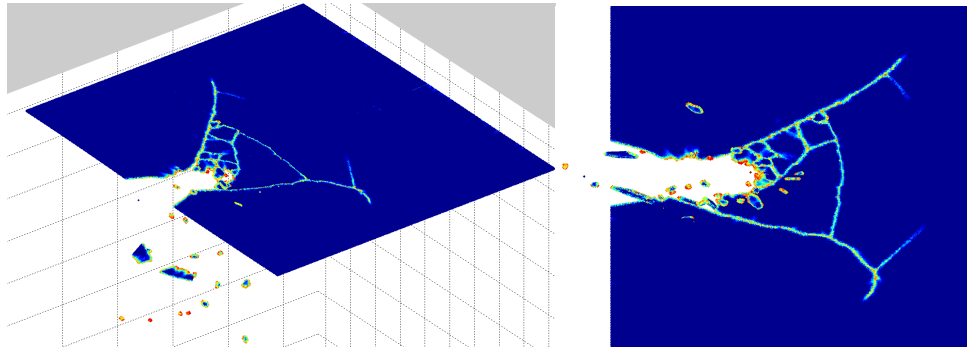


Figure 5.37: Ice fracture at time step of 36000

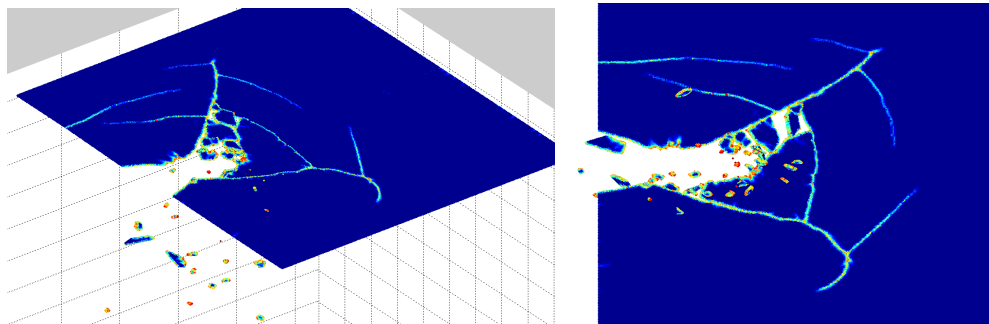


Figure 5.38: Ice fracture at time step of 40000

5.5.4 Conclusion

In this study, peridynamic analysis of the out-of-plane failure of ice is presented with two different models. First model is a 2D model using peridynamic Mindlin plate formulation with water environment represented by Winkler foundation. Second model is a 3D model using peridynamic bond based formulation with water environment represented by a rudimentary body load calculated by Archimedes' law at initial time step. Both models provide promising results by capturing experimentally observed phenomena where ice under out of plane loading fails by circumferential crack propagating from a radial crack. Furthermore, the distinction between finite and semi-infinite floe sizes are classified by the following definitions:

- finite size ice floes are broken by radial crack initiation and propagation

- a semi-infinite ice floe is broken by sequentially forming radial and circumferential cracks

was successfully captured, showing that peridynamic definition of Mindlin plate together with Winkler foundation representation is good enough for future investigations.

5.6 Ice-structure Interaction Applied to the Lighthouse Example

5.6.1 Introduction

Floating ice sheets interacting with rigid vertical structures such as oil platforms represent a significant problem when designing arctic offshore structures. These structures regularly experience high ice loads for prolonged periods of time since ice sheets fail by crushing.

Currently estimation of ice loads on vertical structures is still done by utilizing empirical methods derived from case studies in model-scale or by measuring full-scale ice loading events. As with all estimation methods there is some doubt as to how applicable are those methods when we have varying boundary conditions, ice properties, ice sheet thickness or drift speed. This means that even today, most reliable way of testing a new design is to do a model-scale test to verify the exerted ice loads to structures and that is a time consuming task. This lead to the need for the development of novel numerical methods which could estimate ice loads on structures independent from the boundary conditions, ice properties, ice thickness etc. (Bjerkas et al. (2009)).

One of the biggest issues with numerical ice models is that prevailing mode of ice failure when ice sheet is interacting with a rigid vertical structure is fracture failures. Fracture occurs not only in the contact zone where the prevailing mechanism is crushing but also globally as spalling, buckling, radial and circumferential cracking (Timco (1987)). It can be observed that crack development and propagation has a large influence on the exerted ice forces, which is usually seen as ice load peaks of short duration.

In order to develop model a real life ice-structure interaction model the Norstroms-



Figure 5.39: Norströmsgrund lighthouse (Bjerkas et al. (2009))

grund lighthouse example was used, which is extensively discussed in the literature. In this study peridynamic method is applied in order to calculate ice forces on vertical structures. It will be shown that by using peridynamics as an alternative method, it is possible to produce reliable results alongside a qualitatively comparable crack patterns. There are numerous measured events on the Norströmsgrund lighthouse through the Measurements on Structures in Ice project STRICE (2001-2003). Only a single ice event was considered, which will serve as a basis for comparison to the numerical method presented in this study. Obtained results show that peridynamic formulation is able to capture many of the qualitative observations and also shows comparable ice loads on the lighthouse captured by the real-life measurements (Bjerkas et al. (2009)).

5.6.2 Problem Description

Ice-structure interaction is an important problem when dealing with structures designed for arctic environment that is especially evident for rigid vertical structures such as arctic offshore structures.

For the last several decades the lighthouse Norströmsgrund (Figure 5.39), located in the subarctic Gulf of Bothnia, has served for a full-scale ice load experiment. For this study data from the project STRICE (2001-2003) was used, obtained from Bjerkas

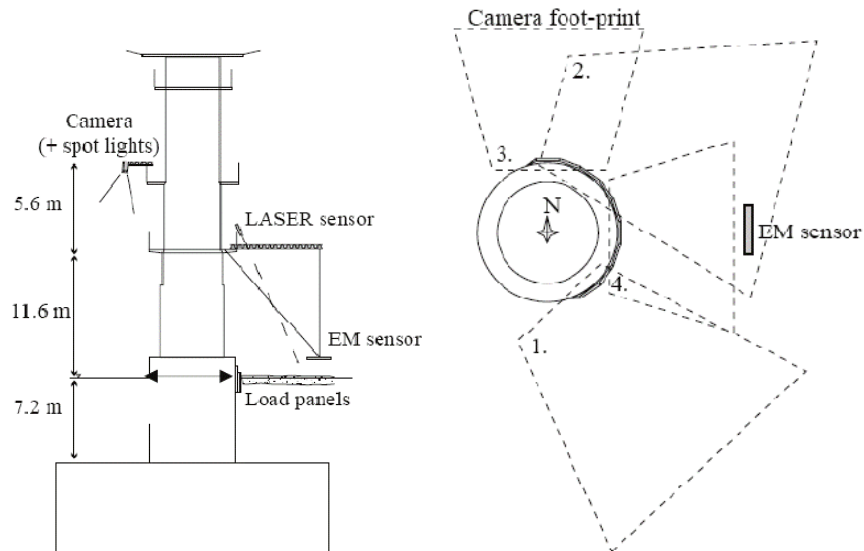


Figure 5.40: Lighthouse Norstrmsgrund full-scale measurement set-up (Bjerkas et al. (2009), Bjerkås (2006))

et al. (2009). Figure 5.40 shows an overview of the experimental setup at Norstrmsgrund during the STRICE project. The structure is made of reinforced concrete where the waterline diameter of the basic structure is 7.2 m. The lighthouse had ice load panels in the waterline, covering 167° of the perimeter from North towards East, where panels had an area of 1.5×1.2 m. Load panel heading towards East was segmented into a matrix of eight sub-segments. Sampling frequency was from 1 Hz to 100 Hz, where frequency was varied dependent on the panel coverage (Bjerkas et al. (2009)).

Event from the morning of March 30th 2003 was selected for peridynamic analyses. The data can be found in Table 1. In this paper the effects of material nonlinearities and friction between ice/structure and ice/water are not taken into account as this requires further development of our current numerical model. The ice is treated as a granular isotropic ice. The effects of loading rate and strain rate on the deformation characteristics for this kind of ice have shown that for unconfined tests under high strain rates ($\dot{\epsilon} > 10^{-3} \text{ s}^{-1}$) ice behaves in a brittle manner (Ralston (1978)).

Table 5.7: Event from the morning of March 30th 2003 (Bjerkas et al. (2009))

	Values
Start	08:46:30
Stop	08:48:00
Ice thickness	0.69 m
Drift speed	0.15 m/s
Drift direction (from)	East
Air temperature	-7.8°C
Max acceleration	0.29 m/s ²
Mean inclination 22 m abs	515 μ rad
Ice condition	Level ice
Ice load sampling	30 Hz

5.6.3 2D Bond Based Peridynamic Model of the Ice Sheet and the Lighthouse

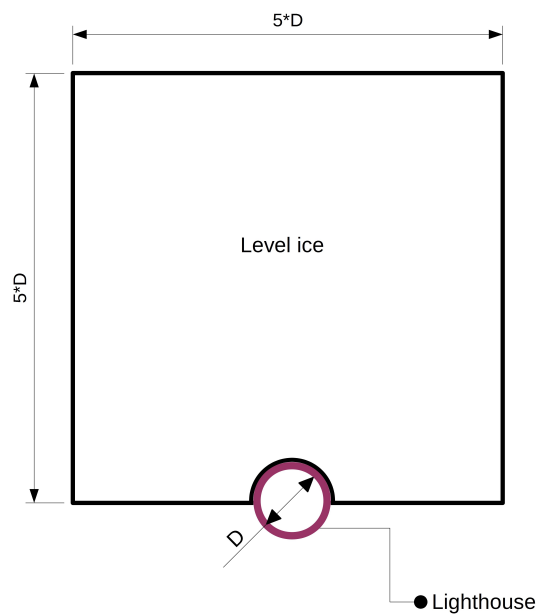


Figure 5.41: Parametrization of the lighthouse model

The lighthouse base diameter is $D = 7.2$ m and ice speed is set to 0.15 m/s. Because the vibrations aren't considered, the lighthouse is modelled as a rigid body. Level ice is a rectangular ice sheet and its length, width and thickness are $L = 36$ m, $W = 36$ m and $T = 0.69$ m, respectively. The edges of the ice sheet, except the one interacting with the lighthouse, are fixed. The configuration of the numerical setup is shown in Figure 5.41. Ice is considered as isotropic and elastic with the following properties shown in Table 5.8.

Table 5.8: Material properties for ice

Values	
Elastic modulus, E	10 GPa
Mass density, ρ	910 kg/m ³
Poisson's ratio, ν	0.3
Strain energy release rate	144 J/m ²

The ice sheet is based on a 2D bond based peridynamic model. The model is solved by using explicit peridynamic solution. The horizon size is chosen as $\delta = 3\Delta x$. The distance between material points is $\Delta x = 0.23$ m, which gives us 24249 material points. According to the principle of time stability, the size of the time step is $\Delta t = 8.0475 \times 10^{-5}$ s. The critical stretch is $s_0 = 1.707 \times 10^{-4}$. It was presumed that the lighthouse is immersed into the level ice sheet in order for the lighthouse to have maximum contact with the ice shelf. Furthermore, ice shelf was fully constrained from three sides except for the edge in contact with the lighthouse.

Figure 5.42 shows the development of the crushing zone and crack development in the ice sheet. The coloration shows damage where higher number means more damage (0 - no damage and 1 - total damage). The cracks are seen to start growing from the ice-structure contact zone, where after reaching a free surface their progress is arrested.

Figure 5.43 shows the integrated surface load on the Norstrmsgrund lighthouse, hence, these loads are equivalent to the global ice loads to the lighthouse model conveyed from the moving ice sheet. As can be seen, the global loads to the structure

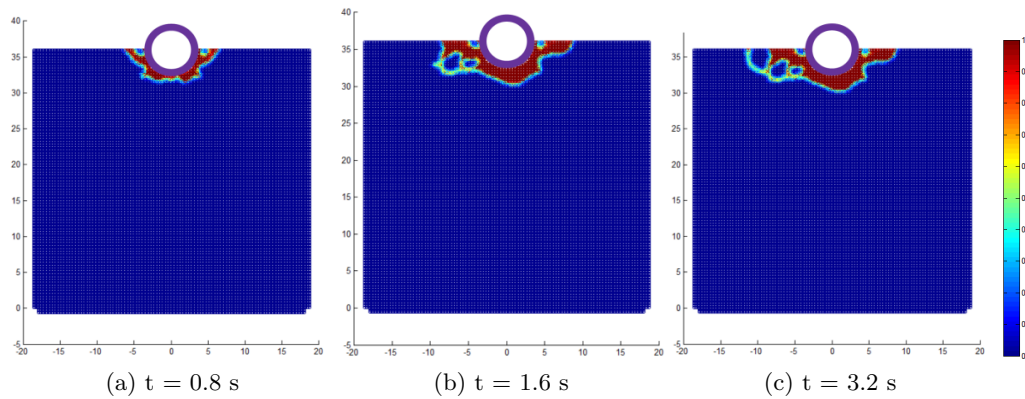


Figure 5.42: Lighthouse Norströmsgrund full-scale measurement set-up (Bjerkas et al. (2009), Bjerkås (2006))

gradually (establishing the contact) increase with time. The peak load coincides with the maximum energy that the ice sheet can sustain. After this point global load is seen to decrease. Furthermore, by comparing results from Figure 5.43 with results obtained by Bjerkas et al. (2009) in Figure 5.44 it can be seen that results from peridynamic simulation have a relatively good agreement with results from FEM analysis for the first 8 seconds.

5.6.4 Conclusion

A peridynamic model of the Norströmsgrund lighthouse and an approaching level ice sheet has been developed. A continuous ice crushing event from the Bjerkas et al. (2009) was adopted and simulated by peridynamics. This study reveals a very interesting aspect of the ice-lighthouse interaction behaviour. First choring zone in front of the lighthouse is developed which is then followed by fracture initiation/propagation in the level ice sheet. Furthermore, results for horizontal global ice force for the first 8 seconds of interaction when compared to FEM results obtained by Bjerkas et al. (2009) show relatively good agreement.

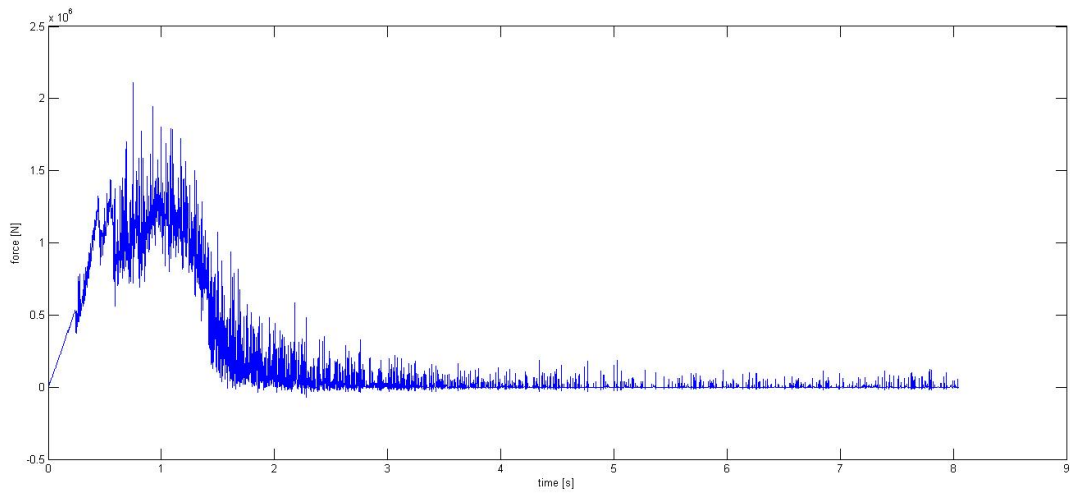


Figure 5.43: Simulated horizontal global ice force on the lighthouse

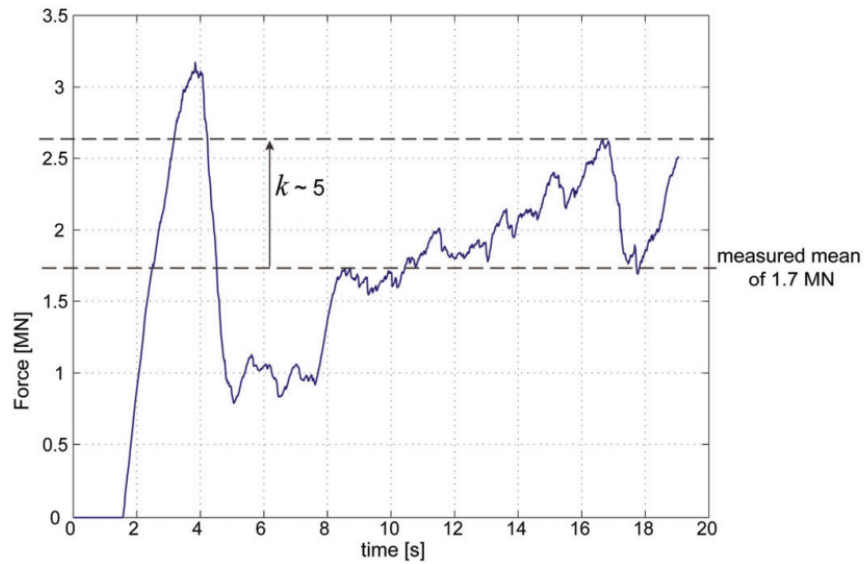


Figure 5.44: Simulated horizontal global ice force on the lighthouse compared to measured loads (Bjerkas et al. (2009))

5.7 Family Member Search Algorithms for Peridynamic Analysis

5.7.1 Introduction

Peridynamic equation of motion is usually solved numerically by using meshless approaches. Family search process is one of the most time consuming parts of a peridynamic analysis. Especially for problems which require continuous update of family members inside the horizon of a material point, the time spent to search for family members becomes crucial. Hence, efficient algorithms are required to reduce the computational time. In this study, various family member search algorithms suitable for peridynamic simulations are presented including brute-force search, region partitioning and tree data structures. By considering problem cases for different number of material points, computational time between different algorithms is compared and the most efficient algorithm is determined.

5.7.2 Comparative Performance of Search Algorithms

In order to compare performance between Brute force search algorithm, Region partitioning search algorithm, Balanced K-d tree, and Boost R-tree with packing algorithm several example cases were considered. Multiple cubic 3-Dimensional PD meshes were created, ranging from 27000 to 8000000 points. Maximum number of family points for the 3D mesh with a horizon of 3Δ is 122. The configuration of the machine used for testing is: *Intel(R) Core(TM) i7-4510U @ 2.0GHz, 8GB RAM, MS Windows 10 x64*. In Table 5.9 timings for family search are presented. Figure 5.45 presents the data from Table 5.9 except the Brute force search since those times aren't comparable with the rest of the algorithms. In Table 5.10 and Figure 5.47, times for building the K-d&R tree structure are shown. Because the Brute force and Region partitioning algorithm don't require any specific structure except for an array of points, there was no need to include them in Table 5.10 and Figure 5.47.

As it can be seen from Table 5.9, Brute Search is the worst performing algorithm as it would be expected. This should indicate that the Brute Search algorithm should

Table 5.9: Family search time comparison

Number of PD points	Brute force search algorithm	Region partitioning search algorithm	Balanced K-d tree	Boost R-tree with packing algorithm
27000	8.428s	0.156s	0.339s	0.289s
64000	57.561s	0.328s	0.776s	0.648s
125000	218.537s	0.797s	1.58s	1.353s
216000	641.56s	1.094s	2.763s	2.393s
343000	1544.48s	1.531s	4.382s	3.841s
512000	3211.89s	2.016s	6.581s	6.229s
729000	6266.8s	2.348s	9.692s	8.421s
1000000	11460.5s	4.28s	13.507s	11.364s
8000000	590930.51s	80.4s	112.498s	94.343s

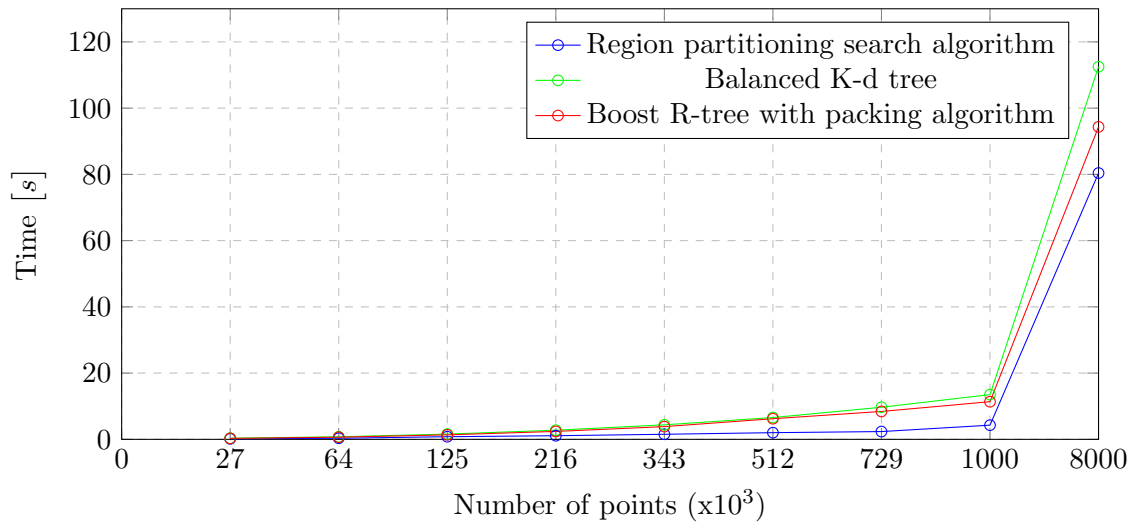


Figure 5.45: Time comparison for different family search algorithms

only be used when doing initial testing of the PD algorithm on relatively small mesh and not as a strategy for complex problems. Comparing the rest of the algorithms, it can be seen that Region partitioning search algorithm performed best. This is further supported when different horizon sizes are used as it can be seen from Figure 5.46 where all tree algorithms are tested for two different horizon sizes $H = 3\Delta$ and 6Δ . Number of family points for horizon sizes 3Δ and 6Δ is 122 and 924, respectively. All of these benefits come with several caveats. First of all, this algorithm is not scalable. As it can be seen from Table 5.9, both R and K-d tree have more or less linear increase of family

search time between 1,000,000 and 8,000,000 point's but Region partitioning algorithm doesn't. Moreover, Region partitioning algorithm is built to fit a specific purpose, which is family search of very regular meshes, preferably rectangular or cubic shaped. Secondly, all of the arrays are either statically allocated or dynamically allocated, but with a purely defined sizes; for example in Region partitioning algorithm size of the family members array for a 3-Dimensional configuration and horizon size of 3Δ is defined as number of points 150 (max size of family members for one point is 122 for this specific horizon). Although this can be easily changed, we would still allocate more memory space than necessary as not all points will have max number of family points (points close to edges of the mesh). Thirdly, if the horizon size changes or horizon shape is not a circle/sphere, user would need to thoroughly rewrite the algorithm which is not easy as the code itself is very complex. This algorithm could be also used for initial testing of the PD algorithm that would require large but symmetrical meshes where use of third party libraries is not possible.

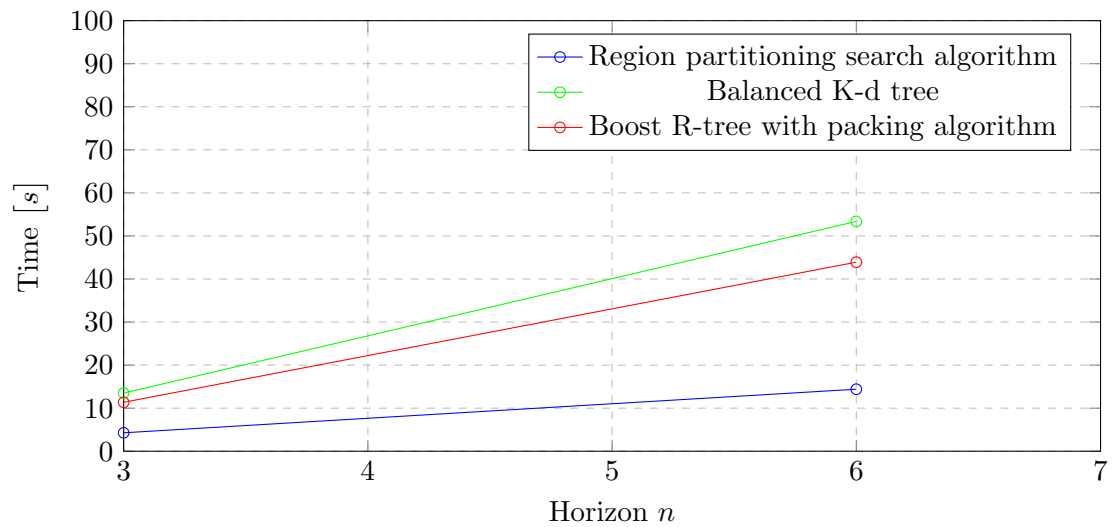


Figure 5.46: Time comparison for different horizon sizes $H = n\Delta$ for 1000000 points

Comparing Balanced K-d tree to the Boost R-tree with packing algorithm, it is obvious from the timings that Boost R-tree performs better. R-tree performs better in building the tree structure and family search (Table 5.10 and Figure 5.47). Furthermore, one of the problems with Balanced K-d tree is relatively high memory consumption

Table 5.10: Times necessary for building the tree structure

Number of PD points	Balanced K-d tree	Boost R-tree with packing algorithm
27000	0.046s	0.014s
64000	0.095s	0.026s
125000	0.186s	0.049s
216000	0.339s	0.1s
343000	0.54s	0.158s
512000	0.782s	0.225s
729000	1.251s	0.327s
1000000	1.62s	0.411s
8000000	17.252s	4.12s

when building the tree structure, compared to the Boost R-tree. The reason behind this is the need for constant sorting of points after each split in order to keep the tree balanced. This makes Boost R-tree more memory friendly for extremely large meshes. Moreover, with the Boost R-tree, it is easy to change the shape of the horizon as the user can overload the geometry definition of the bounding box with different shapes when doing spatial queries. Only possible negative side for the Boost R-tree is dependence on third party development and maintenance of the necessary libraries. In conclusion, Boost R-tree seems currently the best option if there is a need for highly scalable, relatively fast and versatile spatial query algorithm.

5.7.3 Conclusion

In this study, four different family search algorithms were considered including brute-force search, region partitioning search, balanced K-d tree and boost R-tree with packing algorithm. By varying the number of material points inside the solution domain, computational time spent for family member search was determined. According to the results, brute-force search is the worst performing algorithm and it should be used for small number of material points and testing purposes. Although region partitioning search algorithm performed very well, it is limited to large and symmetrical meshes. Finally, it was concluded that Boost R-tree is the best option amongst all four different algorithms considered in this study.

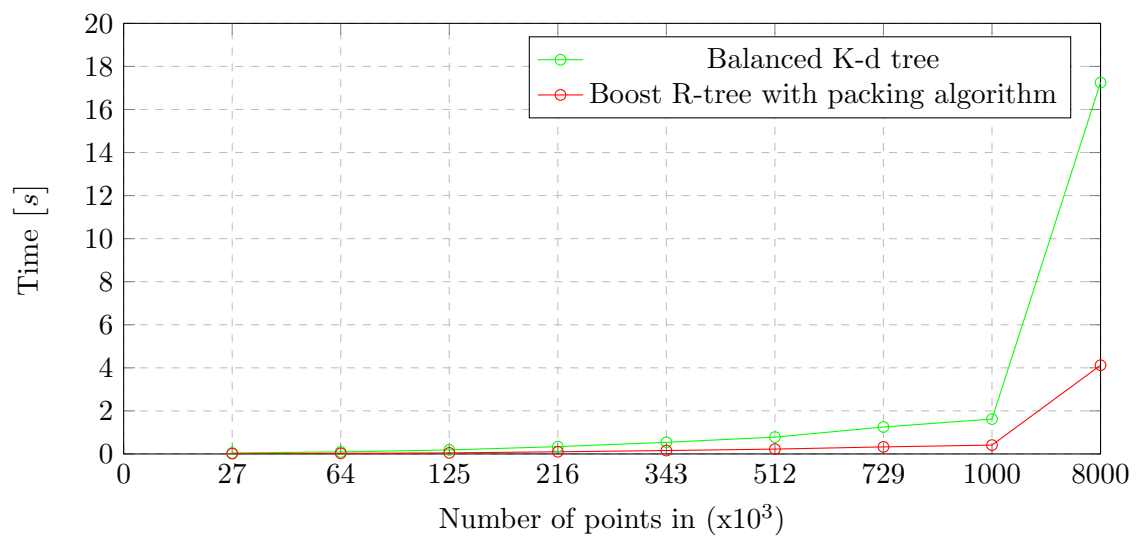


Figure 5.47: Time comparison for building the tree structure between Balanced K-d tree and Boost R-tree with packing algorithm

Chapter 6

Conclusion

It can be seen from previous chapters that Peridynamic (PD) theory has considerable capabilities when it comes to simulating discontinuities, such as cracks, porosity or material interfaces. The main advantage of PD is non-classical (non-local) continuum mechanics formulation, which is very suitable for failure analysis of materials due its mathematical structure. Cracks can occur naturally in the formulation and there is no need to impose an external crack growth law. Furthermore, due to its non-local character, it can capture the phenomenon at multiple scales. Since same type of equations will be used at different scales, a smooth transition is expected. These inherent properties of PD made it applicable in almost every field of solid mechanics research. With this in mind, idea behind this study is to introduce PD into the research field of ice mechanics coupled with marine structures. The main achievements that can be drawn for the present study are:

- Peridynamic solver was developed by using object oriented architecture that allows for easy implementation of new features. The main aim behind solver development was to enable easier modeling of complex ice-structure interaction problems.
- Within the PD solver complete geometry and mesh libraries were developed that enable easy modeling and meshing of complex structures.
- Several state of the art family search algorithms were implemented. Together

with family search algorithms, parallelization of peridynamic code was done in order to reduce computational time.

- Furthermore, rigid body impact library was developed with the aim to solve complex ice-structure interaction problems.
- Mindlin plate resting on Winkler foundation model was developed in order to simulate ice floe in marine environment. The main advantage of this model is its ability to simulate ice floe that is vertically loaded which in return represents realistic model of ice-structure interaction, such as ship colliding with ice sheet.
- Multiple studies were published with an aim to prove that peridynamics is a viable method for solving impact problems

6.0.1 Future Work

Although VOPDSolver was developed within object orientated architecture with loose structure in order to enable future development it still needs to be upgraded to a new version. This is because a lot of new parts were added that weren't taken into account in the initial development. With this in mind as a future work VOPDSolver needs an extensive overhaul from logical and architectural point of view. First thing that needs to be done is to take main parts of the solver, such as geometry, mesh, solvers, loading and constraints and create separate standalone libraries. This will enable even easier future development and maintenance. After this rudimentary user interface needs to be developed. Furthermore, input routines need to be developed to enable transfer of models from 3rd party software, such as Ansys, AutoCad or Abaqus in order to reuse these models for peridynamic analysis.

Winkler's idealization represents the fluid or soil medium as a system of identical but mutually independent, closely spaced, discrete, linearly elastic springs. Due to this idealization, deformation of foundation due to applied load is confined to loaded regions only. If such a foundation is subjected to a partially distributed surface loading, the springs will not be affected beyond the loaded region. This means that model essentially suffers from a complete lack of continuity in the supporting medium. With

this in mind next step should be to extend the formulation to the Pasternak model which assumes existence of shear interaction among the spring elements. This can be accomplished by connecting the ends of the springs to a beam or plate that only undergoes transverse shear deformation. The continuity in this model is characterized by the consideration of the shear layer. Next step should be development of PD fluid model and coupling it with the effects of solid PD model. This can be done as an extension of the work by Demmie and Silling (2007), where they were able to successfully capture gas-solid interaction. Gas-solid interaction could then be extended to the fluid-structure interaction which would be a great improvement from the current Winkler's foundation approximation.

Finally, future efforts should be aimed at introducing physics informed neural network (PINN) into peridynamic framework. Idea behind PINN algorithms is to solve governing partial differential equations (PDE) by minimize its residual (Anitescu et al. (2019)). Although current PINN approach avoids "classical" discretization methods such as FEM or PD as it is using a collocation strategy to discretize the domain it is still a very interesting area that needs further studies.

Bibliography

Adelson-Velskii, G. and E. Landis

. ffan algorithm for the organization of information.”. *Soviet Math*, Pp. 1259–63.

Anitescu, C., E. Atroshchenko, N. Alajlan, and T. Rabczuk

2019. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 59(1):345–359.

Ashton, G. D.

1986. *River and lake ice engineering*. Water Resources Publication.

Attar, M., A. Karrech, and K. Regenauer-Lieb

2014. Free vibration analysis of a cracked shear deformable beam on a two-parameter elastic foundation using a lattice spring model. *Journal of Sound and Vibration*, 333(11):2359–2377.

Ayatollahi, M. and M. Aliha

2009. Analysis of a new specimen for mixed mode fracture tests on brittle materials. *Engineering Fracture Mechanics*, 76(11):1563–1573.

Bayer, R.

1972. Symmetric binary b-trees: Data structure and maintenance algorithms. *Acta informatica*, 1(4):290–306.

Beckmann, N., H.-P. Kriegel, R. Schneider, and B. Seeger

1990. The r*-tree: an efficient and robust access method for points and rectangles.

- In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, Pp. 322–331.
- Bentley, J. L.
1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Benz, W. and E. Asphaug
1995. Simulations of brittle solids using smooth particle hydrodynamics. *Computer physics communications*, 87(1-2):253–265.
- Bird, K. J., R. R. Charpentier, D. L. Gautier, D. W. Houseknecht, T. R. Klett, J. K. Pitman, T. E. Moore, C. J. Schenk, M. E. Tennyson, and C. R. Wandrey
2008. Circum-arctic resource appraisal: Estimates of undiscovered oil and gas north of the arctic circle. Technical report, US Geological Survey.
- Bjerkås, M.
2006. Ice actions on offshore structures—with applications of continuous wavelet transforms on ice load signals.
- Bjerkas, M., A. Guutner, W. Kuhnlein, P. Jochmann, and I. Konuk
2009. Numerical simulation of ice action to a lighthouse. In *Proc. 28th Int. Conf. on Ocean and Artic Engineering, Hawaii, USA*.
- Bobaru, F., M. Yang, L. F. Alves, S. A. Silling, E. Askari, and J. Xu
2009. Convergence, adaptive refinement, and scaling in 1d peridynamics. *International Journal for Numerical Methods in Engineering*, 77(6):852–877.
- Breitenfeld, M., P. H. Geubelle, O. Weckner, and S. Silling
2014. Non-ordinary state-based peridynamic analysis of stationary crack problems. *Computer Methods in Applied Mechanics and Engineering*, 272:233–250.
- Brown, R. A.
2014. Building a balanced kd tree in $o(kn \log n)$ time. *arXiv preprint arXiv:1410.5420*.

Butkovich, T. R.

1956. *Strength studies of sea ice*, volume 20. Snow Ice and Permafrost Research Establishment, Corps of Engineers, US Army.

Chen, H.

2019. A comparison study on peridynamic models using irregular non-uniform spatial discretization. *Computer Methods in Applied Mechanics and Engineering*, 345:539–554.

Chowdhury, S. R., P. Roy, D. Roy, and J. Reddy

2016. A peridynamic theory for linear elastic shells. *International Journal of Solids and Structures*, 84:110–132.

Cole, D., L. Gould, and W. Burch

1985. A system for mounting end caps on ice specimens. *Journal of Glaciology*, 31(109):362–365.

Cole, D. M.

2001. The microstructure of ice and its influence on mechanical properties. *Engineering Fracture Mechanics*, 68(17-18):1797–1822.

De Meo, D., C. Diyaroglu, N. Zhu, E. Oterkus, and M. A. Siddiq

2016. Modelling of stress-corrosion cracking by using peridynamics. *International Journal of Hydrogen Energy*, 41(15):6593–6609.

Demmie, P. and S. Silling

2007. An approach to modeling extreme loading of structures using peridynamics. *Journal of Mechanics of Materials and Structures*, 2(10):1921–1945.

Dempsey, J., R. Adamson, and S. Mulmule

1999. Scale effects on the in-situ tensile strength and fracture of ice. part ii: First-year sea ice at resolute, nwt. *International journal of fracture*, 95(1-4):347.

Di Paola, M., F. Marino, and M. Zingales

2009. A generalized model of elastic foundation based on long-range interactions:

- Integral and fractional model. *International Journal of Solids and Structures*, 46(17):3124–3137.
- Dipasquale, D., M. Zaccariotto, and U. Galvanetto
2014. Crack propagation with adaptive grid refinement in 2d peridynamics. *International Journal of Fracture*, 190(1-2):1–22.
- Diyaroglu, C.
2016. *Peridynamics and its applications in marine structures*. PhD thesis, University of Strathclyde.
- Diyaroglu, C., E. Oterkus, S. Oterkus, and E. Madenci
2015. Peridynamics for bending of beams and plates with transverse shear deformation. *International Journal of Solids and Structures*, 69:152–168.
- Domínguez, J. M., A. J. Crespo, M. Gómez-Gesteira, and J. C. Marongiu
2011. Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 67(12):2026–2042.
- Dykins, J.
1970. Ice engineering-tensile properties of sea ice grown in a confined system. Technical report, NAVAL CIVIL ENGINEERING LAB PORT HUENEME CA.
- Dykins, J.
1971. Ice engineering material properties for a limited range of conditions. In *Technical Report R720*. US Navy Civil Engineering Laboratory Port Hueneme, CA.
- Elruby, A., S. Nakhla, and A. Hussein
2018. Automating xfm modeling process for optimal failure predictions. *Mathematical Problems in Engineering*, 2018.
- Emmrich, E. and O. Weckner
2007. The peridynamic equation and its spatial discretisation. *Mathematical Modelling and Analysis*, 12(1):17–27.

- Farjoo, M., W. Daniel, and P. A. Meehan
2012. Modelling a squat form crack on a rail laid on an elastic foundation. *Engineering Fracture Mechanics*, 85:47–58.
- Foster, J. T., S. A. Silling, and W. W. Chen
2010. Viscoplasticity using peridynamics. *International journal for numerical methods in engineering*, 81(10):1242–1258.
- Frederking, R. M. and G. W. Timco
1984. Measurement of shear strength of granular/discontinuous-columnar sea ice. *Cold Regions Science and Technology*, 9(3):215–220.
- Frederking, R. M. and G. W. Timco
1986. Field measurements of the shear strength of columnar-grained sea ice.
- Freimanis, A. and A. Paeglitis
2017. Mesh sensitivity in peridynamic quasi-static simulations. *Procedia Engineering*, 172:284–291.
- Gammon, P., H. Kiefte, and M. Clouter
1983. Elastic constants of ice samples by brillouin spectroscopy. *The Journal of Physical Chemistry*, 87(21):4025–4029.
- Goldstine, H. and J. von Neumann
1963. Coding of some combinatorial (sorting) problems. *John von Neumann Collected Works: Design of Computers, Theory of Automata and Numerical Analysis*, 5:196–214.
- Grape, J. A., E. M. Schulson, et al.
1992. Effect of confining stress on brittle indentation failure of columnar ice. *International Journal of Offshore and Polar Engineering*, 2(03).
- Greene, D.
1989. An implementation and performance analysis of spatial data access methods.

In [1989] *Proceedings. Fifth International Conference on Data Engineering*, Pp. 606–615. IEEE.

Griffith, A. A.

1921. Vi. the phenomena of rupture and flow in solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 221(582-593):163–198.

Griffiths, D. and G. G. Mustoe

2001. Modelling of elastic continua using a grillage of structural elements based on discrete element concepts. *International Journal for Numerical Methods in Engineering*, 50(7):1759–1775.

Guibas, L. J. and R. Sedgewick

1978. A dichromatic framework for balanced trees. In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, Pp. 8–21. IEEE.

Guttman, A.

1984. *R-trees: A dynamic index structure for spatial searching*, volume 14. ACM.

Ha, Y. D. and F. Bobaru

2010. Studies of dynamic crack propagation and crack branching with peridynamics. *International Journal of Fracture*, 162(1-2):229–244.

Hawkes, I. and M. Mellor

1972. Deformation and fracture of ice under uniaxial stress. *Journal of Glaciology*, 11(61):103–131.

Hill, B. T.

2009. Ship iceberg collisions database.

Hillerborg, A., M. Mod er, and P.-E. Petersson

1976. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and concrete research*, 6(6):773–781.

Hoare, C. A.

1962. Quicksort. *The Computer Journal*, 5(1):10–16.

Howard, M. P., J. A. Anderson, A. Nikoubashman, S. C. Glotzer, and A. Z. Panagiotopoulos

2016. Efficient neighbor list calculation for molecular simulation of colloidal systems using graphics processing units. *Computer Physics Communications*, 203:45–52.

Javili, A., R. Morasata, E. Oterkus, and S. Oterkus

2019. Peridynamics review. *Mathematics and Mechanics of Solids*, 24(11):3714–3739.

Kerr, A. D.

1976. The bearing capacity of floating ice plates subjected to static or quasi-static loads. *Journal of glaciology*, 17(76):229–268.

Kilic, B. and E. Madenci

2010. An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. *Theoretical and Applied Fracture Mechanics*, 53(3):194–204.

Kuehn, G., R. Lee, W. Nixon, and E. Schulson

1990. The structure and tensile behavior of first-year sea ice and laboratory-grown saline ice.

Lange, M. A. and T. J. Ahrens

1983. The dynamic tensile strength of ice and ice-silicate mixtures. *Journal of Geophysical Research: Solid Earth*, 88(B2):1197–1208.

Langhorne, P., K. Stone, and C. Smith

1999. The bearing capacity of saline ice sheets: centrifugal modelling. *Canadian geotechnical journal*, 36(3):467–481.

Le, Q. and F. Bobaru

2018. Surface corrections for peridynamic models in elasticity and fracture. *Computational Mechanics*, 61(4):499–518.

- Libraries, B. C.
2009. Spatial indexes introduction.
- Liu, R., Y. Xue, X. Lu, and W. Cheng
2018. Simulation of ship navigation in ice rubble based on peridynamics. *Ocean Engineering*, 148:286–298.
- Lu, W., R. Lubbad, and S. Løset
2015a. In-plane fracture of an ice floe: A theoretical study on the splitting failure mode. *Cold Regions Science and Technology*, 110:77–101.
- Lu, W., R. Lubbad, and S. Løset
2015b. Out-of-plane failure of an ice floe: radial-crack-initiation-controlled fracture. *Cold Regions Science and Technology*, 119:183–203.
- Lubbad, R. and S. Løset
2011. A numerical model for real-time simulation of ship–ice interaction. *Cold Regions Science and Technology*, 65(2):111–127.
- Macek, R. W. and S. A. Silling
2007. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design*, 43(15):1169–1178.
- Madenci, E., A. Barut, and M. Dorduncu
2019. *Peridynamic differential operator for numerical analysis*. Springer.
- Madenci, E. and E. Oterkus
2014. Peridynamic theory. In *Peridynamic Theory and Its Applications*, Pp. 19–43. Springer.
- Madenci, E. and S. Oterkus
2016. 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.

Madenci, E. and S. Oterkus

2017. Ordinary state-based peridynamics for thermoviscoelastic deformation. *Engineering Fracture Mechanics*, 175:31–45.

Maeno, N. and T. Ebinuma

1983. Pressure sintering of ice and its implication to the densification of snow at polar glaciers and ice sheets. *The Journal of Physical Chemistry*, 87(21):4103–4110.

Mattson, W. and B. M. Rice

1999. Near-neighbor calculations using a modified cell-linked list method. *Computer Physics Communications*, 119(2-3):135–148.

Matysiak, S. and V. Pauk

2003. Edge crack in an elastic layer resting on winkler foundation. *Engineering fracture mechanics*, 70(17):2353–2361.

Mellor, M.

1986. Mechanical behavior of sea ice. In *The geophysics of sea ice*, Pp. 165–281. Springer.

Menge, J. R. and K. Jones

1993. The tensile strength of first-year sea ice. *Journal of Glaciology*, 39(133):609–618.

Michel, B.

1978a. Ice mechanics.

Michel, B.

1978b. The strength of polycrystalline ice. *Canadian Journal of Civil Engineering*, 5(3):285–300.

Michel, B. and R. Ramseier

1971. Classification of river and lake ice. *Canadian Geotechnical Journal*, 8(1):36–45.

- Moës, N. and T. Belytschko
2002. Extended finite element method for cohesive crack growth. *Engineering fracture mechanics*, 69(7):813–833.
- Morrison, C. N., M. Zhang, D. Liu, and A. P. Jivkov
2015. Site-bond lattice modelling of damage process in nuclear graphite under bending.
- Mulmule, S. and J. Dempsey
1999. Scale effects on sea ice fracture. *Mechanics of Cohesive-frictional Materials: An International Journal on Experiments, Modelling and Computation of Materials and Structures*, 4(6):505–524.
- Nanthikesan, S. and S. S. Sunder
1994. Anisotropic elasticity of polycrystalline ice. *Cold regions science and technology*, 22(2):149–169.
- Nevel, D. E.
1965. A semi-infinite plate on an elastic foundation. Technical report, COLD REGIONS RESEARCH AND ENGINEERING LAB HANOVER NH.
- Nixon, W. A. and E. Schulson
1988. The fracture toughness of ice over a range of grain sizes. *Journal of Offshore Mechanics and Arctic Engineering*, 110(2):192–196.
- Nobili, A., E. Radi, and L. Lanzoni
2014. A cracked infinite kirchhoff plate supported by a two-parameter elastic foundation. *Journal of the European Ceramic Society*, 34(11):2737–2744.
- Notz, D.
2005. *Thermodynamic and fluid-dynamical processes in sea ice*. PhD thesis, University of Cambridge Cambridge.
- O’Brien, G. S. and C. J. Bean

2011. An irregular lattice method for elastic wave propagation. *Geophysical Journal International*, 187(3):1699–1707.
- O’Grady, J. and J. Foster
2014. Peridynamic beams: a non-ordinary, state-based model. *International Journal of Solids and Structures*, 51(18):3177–3183.
- Oterkus, E., A. Barut, and E. Madenci
2010. Damage growth prediction from loaded composite fastener holes by using peridynamic theory. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th AIAA/ASME/AHS Adaptive Structures Conference 12th*, P. 3026.
- Oterkus, E., I. Guven, and E. Madenci
2012. Impact damage assessment by using peridynamic theory. *Open Engineering*, 2(4):523–531.
- Oterkus, E. and E. Madenci
2012. Peridynamics for failure prediction in composites. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, P. 1692.
- Oterkus, S., J. Fox, and E. Madenci
2013. Simulation of electro-migration through peridynamics. In *2013 IEEE 63rd Electronic Components and Technology Conference*, Pp. 1488–1493. IEEE.
- Oterkus, S. and E. Madenci
2015. Peridynamics for antiplane shear and torsional deformations. *Journal of Mechanics of Materials and Structures*, 10(2):167–193.
- Oterkus, S. and E. Madenci
2017. Peridynamic modeling of fuel pellet cracking. *Engineering Fracture Mechanics*, 176:23–37.

- Oterkus, S., E. Madenci, and A. Agwai
2014. Peridynamic thermal diffusion. *Journal of Computational Physics*, 265:71–96.
- Oterkus, S., E. Madenci, and E. Oterkus
2017. Fully coupled poroelastic peridynamic formulation for fluid-filled fractures. *Engineering geology*, 225:19–28.
- Paige, R. A. and C. W. Lee
1967. Preliminary studies on sea ice in mcmurdo sound, antarctica, during deep freeze 65. *Journal of Glaciology*, 6(46):515–528.
- Pazdniakou, A. and P. Adler
2012. Lattice spring models. *Transport in porous media*, 93(2):243–262.
- Perré, P., G. Almeida, M. Ayouz, and X. Frank
2016. New modelling approaches to predict wood properties from its cellular structure: image-based representation and meshless methods. *Annals of forest science*, 73(1):147–162.
- Petrovic, J.
2003. Review mechanical properties of ice and snow. *Journal of materials science*, 38(1):1–6.
- Rabczuk, T. and T. Belytschko
2004. Cracking particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering*, 61(13):2316–2343.
- Rabczuk, T. and T. Belytschko
2007. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer methods in applied mechanics and engineering*, 196(29-30):2777–2799.
- Ralston, T.
1978. An analysis of ice sheet indentation. In *Fourth International Symposium on*

- Ice Problems*, Pp. 13–31. International Association for Hydraulic Research Luleå, Sweden.
- Reddy, J., A. Srinivasa, A. Arbind, and P. Khodabakhshi
2013. On gradient elasticity and discrete peridynamics with applications to beams and plates. In *Advanced Materials Research*, volume 745, Pp. 145–154. Trans Tech Publ.
- Rist, M., P. Sammonds, S. Murrell, P. Meredith, C. Doake, H. Oerter, and K. Matsuki
1999. Experimental and theoretical fracture mechanics applied to antarctic ice fracture and surface crevassing. *Journal of Geophysical Research: Solid Earth*, 104(B2):2973–2987.
- Rist, M., P. Sammonds, H. Oerter, and C. Doake
2002. Fracture of antarctic shelf ice. *Journal of Geophysical Research: Solid Earth*, 107(B1):ECV–2.
- Rubinstein, A. A.
1985. Macrocrack interaction with semi-infinite microcrack array. *International Journal of Fracture*, 27(2):113–119.
- Sand, B.
2008. *Nonlinear finite element simulations of ice forces on offshore structures*. PhD thesis, Luleå tekniska universitet.
- Schulson, E., P. Lim, and R. Lee
1984. A brittle to ductile transition in ice under tension. *Philosophical Magazine A*, 49(3):353–363.
- Schulson, E. M.
2001. Brittle failure of ice. *Engineering fracture mechanics*, 68(17-18):1839–1887.
- Schulson, E. M. and P. Duval
2009. *Creep and fracture of ice*. Cambridge University Press.

Seleson, P. and D. J. Littlewood

2016. Convergence studies in meshfree peridynamic simulations. *Computers & Mathematics with Applications*, 71(11):2432–2448.

Silling, S.

2004. Emu users manual, code ver. 2.6 d. *Sandia National Laboratories, Albuquerque*.

Silling, S. A.

2000. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175–209.

Silling, S. A. and E. Askari

2004. Peridynamic modeling of impact damage. In *ASME/JSME 2004 pressure vessels and piping conference*, Pp. 197–205. American Society of Mechanical Engineers Digital Collection.

Silling, S. A. and E. Askari

2005. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17-18):1526–1535.

Silling, S. A. and F. Bobaru

2005. Peridynamic modeling of membranes and fibers. *International Journal of Non-Linear Mechanics*, 40(2-3):395–409.

Silling, S. A., M. Epton, O. Weckner, J. Xu, and E. Askari

2007. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184.

Sodhi, D. S.

1995. Breakthrough loads of floating ice sheets. *Journal of cold regions engineering*, 9(1):4–22.

Squire, V. A., R. J. Hosking, A. D. Kerr, and P. J. Langhorne

2012. *Moving loads on ice plates*, volume 45. Springer Science & Business Media.

- Sukumar, N., N. Moës, B. Moran, and T. Belytschko
2000. Extended finite element method for three-dimensional crack modelling. *International journal for numerical methods in engineering*, 48(11):1549–1570.
- Taylor, M. and D. J. Steigmann
2015. A two-dimensional peridynamic model for thin plates. *Mathematics and Mechanics of Solids*, 20(8):998–1010.
- Timco, G.
1987. Indentation and penetration of edge-loaded freshwater ice sheets in the brittle range. *Journal of Offshore Mechanics and Arctic Engineering*, 109(3):287–294.
- Timco, G. and R. Frederking
1990. Compressive strength of sea ice sheets. *Cold Regions Science and Technology*, 17(3):227–240.
- Timco, G. and R. Frederking
1991. Seasonal compressive strength of beaufort sea ice sheets. In *Ice-Structure Interaction*, Pp. 267–282. Springer.
- Timco, G. and W. Weeks
2010. A review of the engineering properties of sea ice. *Cold regions science and technology*, 60(2):107–129.
- Underwood, P.
1983. Dynamic relaxation. *Computational method for transient analysis*, 1:245–263.
- Vazic, B., C. Diyaroglu, E. Oterkus, and S. Oterkus
2019a. Family member search algorithms for peridynamic analysis. *Journal of Peridynamics and Nonlocal Modeling*, Pp. 1–26.
- Vazic, B., E. Oterkus, and S. Oterkus
2019b. Peridynamic approach for modelling ice-structure interactions. In *Trends in the Analysis and Design of Marine Structures: Proceedings of the 7th International*

Conference on Marine Structures (MARSTRUCT 2019, Dubrovnik, Croatia, 6-8 May 2019), P. 55. CRC Press.

Vazic, B., E. Oterkus, and S. Oterkus

2019c. Peridynamic model for a mindlin plate resting on a winkler elastic foundation. *Journal of Peridynamics and Nonlocal Modeling*.

Vazic, B., H. Wang, C. Diyaroglu, S. Oterkus, and E. Oterkus

2017. Dynamic propagation of a macrocrack interacting with parallel small cracks. *AIMS Materials Science*, 4(1):118–136.

Viccione, G., V. Bovolín, and E. P. Carratelli

2008. Defining and optimizing algorithms for neighbouring particle identification in sph fluid simulations. *International Journal for Numerical Methods in Fluids*, 58(6):625–638.

Wadhams, P.

2003. How does arctic sea ice form and decay. *Retrieved May, 19(2008):275–332*.

Wald, I. and V. Havran

2006. On building fast kd-trees for ray tracing, and on doing that in $O(n \log n)$. In *2006 IEEE Symposium on Interactive Ray Tracing*, Pp. 61–69. IEEE.

Wang, H., Z. Liu, D. Xu, Q. Zeng, and Z. Zhuang

2016. Extended finite element method analysis for shielding and amplification effect of a main crack interacted with a group of nearby parallel microcracks. *International Journal of Damage Mechanics*, 25(1):4–25.

Weeks, W. F. and S. F. Ackley

1986. The growth, structure, and properties of sea ice. In *The geophysics of sea ice*, Pp. 9–164. Springer.

Winkler, E.

. Die lehre von der elastizitat und festigkeit. 1867. *Dominicus, Prague*.

Xu, X.-P. and A. Needleman

1994. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42(9):1397–1434.