

NUMERICAL INVESTIGATION OF FRACTURE OF POLYCRYSTALLINE ICE UNDER
DYNAMIC LOADING

by

© Igor Gribanov

A thesis submitted to the School of Graduate Studies in
partial fulfillment of the requirement for the degree of
Doctor of Philosophy

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

May 2020

St. John's, Newfoundland, Canada

Abstract

Cohesive zone model is a promising technique for simulating fracture processes in brittle ice. In this work it is applied to investigate the fracture behavior of polycrystalline cylindrical samples under uniaxial loading conditions, four-point beam bending, and L-shaped beam bending. In each case, the simulation results are compared with the corresponding experimental data that was collected by other researchers. The model is based on the implicit finite element method combined with Park-Paulino-Roesler formulation for cohesive potential and includes an adaptive time stepping scheme, which takes into account the rate of damage and failure of cohesive zones. The benefit of the implicit scheme is that it allows larger time steps than explicit integration. Material properties and model parameters are calibrated using available experimental data for freshwater ice and sea ice samples.

For polycrystalline ice, granular geometry is generated and cohesive zones are inserted between grains. Simulations are performed for samples with different grain sizes, and the resulting stress-strain and damage accumulation curves are recorded. Investigation of the dependency between the grain size and fracture strength shows a strengthening effect that is consistent with experimental results.

The proposed framework is also applied to simulate the dynamic fracture processes in L-shaped beams of sea ice, in which case the cohesive zones are inserted between the elements of the mesh. Evolution of the stress distribution on the surface of the beam is modeled for the duration of the loading process, showing how it changes with progressive accumulation of damage in the material, as well as the development of cracks. An analytical formula is derived for estimating the breaking force based on the dimensions of the beam and the ice strength. Experimental data obtained from the 2014-2016 tests are re-evaluated with the aid of this new analysis.

The computation is implemented efficiently with GPU acceleration, allowing to handle geometries with higher resolution than would be possible otherwise. Several technical contributions are described in detail including GPU-accelerated FEM implementation, an efficient way of creation of sparse matrix structure, and comparison of different unloading/reloading relations when using an implicit integration scheme. A mechanism for collision response allows modeling the interaction of fragmented material. To evaluate the collision forces, an algorithm for computing first and second point-triangle distance derivatives was developed. The source code is made available as open-source.

Acknowledgements

This thesis is the result of my academic work at Memorial University of Newfoundland. I am grateful for this meaningful experience and for the extraordinary opportunity. While working on this project I encountered many successes and failures, and sometimes the seemingly minor obstacles were difficult to overcome. Hopefully, some of the presented ideas and technical solutions will be useful to other researchers and students.

I want to thank my research supervisor Professor Rocky Taylor for his guidance and support. This thesis would not have been possible without his helpful direction, advice, and insight. He is one of the most hard-working researchers I ever met and has excellent organizational skills and a charismatic personality. I also owe a debt of gratitude to my co-supervisor Dr. Robert Sarracino for valuable discussions and encouragement. I also want to thank my co-supervisor, Dr. Michael Slawinski, conversations with whom were insightful.

I would like to thank Dr. Alexander Kupershtokh, Dr. Denis Karpov and Dr. Dmitriy Medvedev from Lavrentyev Institute of Hydrodynamics for introducing me to CUDA programming. Special gratitude goes to my lifelong friend Elena Fominykh.

I want to thank my friends and colleagues who have contributed to this work via insightful discussions: Mohamed Aly, Ridwan Hossain Razin, Marjan Boroojerdi, Hamid Shayanfar, Soroosh Afzali, Reza Zeinali, Renat Yulmetov, Zachary Koszegi-Faulkner, James C. Lim, and Kashfi Habib.

The financial support was made available by Memorial University of Newfoundland, the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Research Development Corporation of Newfoundland and Labrador (RDC), Hibernia Management and Development Company, Ltd. (HMDC), Terra Nova Development (Suncor Energy Inc. - Operator) and InnovateNL. NVIDIA provided a GPU via their grant program.

This thesis is dedicated to my dearest friend Anton Mikov.

Contents

List of Tables	IX
List of Figures	XI
1 Introduction and Overview	1
1.1 Background and motivation	1
1.2 Approaches to ice modeling	3
1.3 Modeling large deformation and fracture	11
1.3.1 Fracture methods in computer graphics	12
1.3.2 Optimization for a 2D case	14
1.3.3 Modeling with cohesive zones	15
1.4 Research method and scope of work	17
1.5 Theoretical background	19
1.5.1 Deformation and strain	19
1.5.2 Strain energy	22
1.5.3 Cauchy momentum equation	24
1.5.4 Finite element method	24
1.5.5 Cohesive zones and grains within FEM	31
1.5.6 Collision detection and response	32

1.6	Organization of thesis	34
	References	35
2	Review of Ice Mechanics	43
2.1	Crystal structure, microstructure, types of ice and formation	43
2.1.1	Crystal structure and microstructure	43
2.1.2	Types of ice	46
2.1.3	Formation of sea ice	52
2.2	Continuum behavior, elasticity, creep and constitutive models	55
2.3	Fracture, strength, flaws, and ice-structure interaction	61
2.4	Experimental techniques for measurement of ice properties	66
2.4.1	Small-scale tests	67
2.4.2	Full-scale observations	70
2.4.3	Medium-scale tests	73
2.5	High-pressure zones, the mechanics of ice-structure interaction	76
2.6	Probabilistic analysis	78
2.6.1	Pressure-area relationship	80
2.6.2	Distributions for hull pressures	82
2.6.3	Other applications of probabilistic analysis	85
2.7	Other approaches to describing fracture	85
2.7.1	Wing crack models	86
2.7.2	Linear elastic fracture mechanics	88

2.7.3	Energy-based approach to understanding the fracture process of ice	91
2.7.4	Floe impact on a structure	92
	References	94
3	Cohesive Zone Micromechanical Model for Compressive and Tensile Failure of Polycrystalline Ice	110
3.1	Introduction	112
3.1.1	Related work	115
3.1.2	Our contribution	117
3.2	Background	118
3.3	Simulation overview	122
3.3.1	Geometry of deformable samples	123
3.3.2	Integration and discretization schemes	125
3.3.3	Elastic forces	126
3.3.4	Cohesive zone model	127
3.3.5	Collision response	131
3.3.6	Time step selection	132
3.4	Numerical results	133
3.5	Conclusions and future work	141
	References	142
4	Parallel Implementation of Implicit Finite Element Model with Cohesive Zones and Collision Response using CUDA	148
4.1	Introduction	150

4.1.1	Related work on parallel implementation of FEM	151
4.1.2	Suggested approach	155
4.2	Model formulation	156
4.2.1	Discretized equation of motion and Newmark-beta method	157
4.2.2	Elastic response	160
4.2.3	Forces from cohesive elements	162
4.2.4	Unloading/reloading of CZ	165
4.2.5	Collision response	167
4.3	Implementation	169
4.3.1	Sparse matrix format	171
4.3.2	GPU-specific optimizations	173
4.3.3	Implementation of collision detection and response	175
4.3.4	Detection of failed cohesive zones	176
4.4	Simulation results and benchmarking	177
4.4.1	Comparison of the unloading/reloading relations	179
4.4.2	Comparison between brittle and ductile CZs	185
4.4.3	Performance analysis	187
4.5	Concluding remarks	189
	References	191

5 The Gradient and the Hessian of the Distance Between Point and Triangle in 3D 196

5.1	Introduction	198
-----	------------------------	-----

5.2	Mathematical formulation	200
5.3	Point-edge and point-point cases	210
5.4	Algorithm and testing	211
5.4.1	Discussion	213
5.5	Conclusion	215
	References	215
6	Application of Cohesive Zone Model to the Fracture Process of Freshwater Polycrystalline Ice under Flexural Loading	217
6.1	Introduction	218
6.1.1	Related work on modeling	220
6.2	Description of the method	221
6.3	Results and discussion	224
6.4	Conclusion	228
	References	229
7	Investigation of Mixed Mode Fracture of L-shaped Sea Ice Beams	234
7.1	Introduction	236
7.2	Cohesive zone modeling	238
7.3	Analytical approach	243
7.3.1	Comparison of the analytical solution with FEM parametric tests . . .	247
7.3.2	Application of the analytical model to the field test data	250
7.4	Discussion and conclusions	251

References	253
8 Summary and Conclusion	255
8.1 Summary	255
8.2 Discussion	259
8.3 Recommendations for future work	261
References	262
A Appendix: Reference Implementation for Calculating the Gradient and the Hessian of the Distance between a Point and a Triangle in 3D	264

List of Tables

2.1	Categories and subcategories of freshwater and seawater ice types.	47
2.2	Data sources.	65
3.1	Properties of the material.	128
3.2	Model parameters.	128
3.3	The modeled samples with different grain arrangements, i.e. different randomly generated geometries. The 4000-grain sample was only simulated in compressive test.	134
3.4	Values of material properties.	135
3.5	Values of simulation parameters.	135
4.1	Formulation of uncoupled and thermodynamically consistent unloading/reloading relations.	166
4.2	Materials used in the simulation.	178
4.4	Mesh parameters of the cylindrical sample for compressive tests.	179
4.6	Running time of the simulations.	188
4.7	Computation steps.	188
5.1	Relative errors for squared distance and its first and second derivatives. The maximum values from 10 million test cases are shown.	213
6.1	Parameters of the material.	224
7.1	Parameters of cohesive zones and elastic elements.	240

7.2	Series of parametric tests.	247
7.3	Calculated flexural/shear strength for experimentally obtained forces.	251

List of Figures

1.1	Pack ice event on March 30, 2017. Near St.John's, Newfoundland.	8
1.2	Fracture of thin sheets.	14
1.3	Initial and final configurations of a deformable object.	20
1.4	Deformation of a 2D infinitesimal element.	21
1.5	A single tetrahedral element.	25
1.6	A plane defined by $\zeta_1 = 1/3$	27
1.7	Schematic representation of a cohesive zone.	32
1.8	Levels of the bounding volume hierarchy going from (a) the top of the tree to (d) the leaves of the tree.	33
1.9	Grains rest on the top of a solid disk. Areas of contact are shown in red color.	34
2.1	Crystal structure of the ice Ih.	44
2.2	Movement of dislocation through the ice crystal.	45
2.3	Structure of first year ice.	54
2.4	Plastic deformations in three-point bending of pure crystals with different orientations of c-axes.	56
2.5	A typical strain-time curve for a constant strain stress test.	56
2.6	A spring-and-dashpot model of viscoelastic behavior of ice under uniaxial stress.	58
2.7	Schematic diagram of reversible grain boundary sliding.	59
2.8	Fracture modes.	62

2.9	Deformation mode map of ice failure as a function of indentation rate and aspect ratio.	63
2.10	(a) Extraction of a core sample from the frozen cover; (b) Fractured samples after testing; (c) Bubbles and other inclusions in a sample.	67
2.11	Schematic representation of the AFPB machine.	68
2.12	Schematic of the large indentation apparatus in the trench.	74
2.13	Illustration of spalling, crushing, extrusion and formation of hpzs.	77
2.14	Theoretical pressure-area curve constrained to the observed data.	81
2.15	Pressure vs. area for ship-ice interaction.	81
2.16	Comparison of local pressure for impact events on the bow shoulder measured in 2013 and 2014 field tests.	83
2.17	Formation of wing crack in a brittle solid under compression.	87
2.18	A cracked structure with initial notch of length a under stress σ	88
2.19	Fracture modes.	90
2.20	(a) Imperfect contact between an ice floe and a structure wall, (b) problem idealization, (c) detailed view and definitions.	93
3.1	Schematic illustration of spalling, extrusion and high-pressure zone formation.	118
3.2	Stress-strain curves for granular freshwater ice Ih at -10°	120
3.3	Fracture strength versus inverse square root of grain size for sample strained under (a) compression and (b) tension. The strain rate is 10^{-3} s^{-1} in both cases. Reproduced after Lee and Schulson.	121
3.4	Cylinder with 1000 grains. Top and bottom caps do not contain cohesive zones.	123
3.5	Two grains connected with cohesive zones.	124
3.6	Distribution of grain sizes in a cylindrical sample with 1250 grains.	125

3.7	Traction-separation relations for (a) normal opening, and (b) tangential opening with parameter values ($\phi_n = \phi_t = 30$, $\sigma_{max} = 4 \times 10^5$, $\tau_{max} = 1.5 \times 10^6$, $\alpha = \beta = 3$, $\lambda_n = \lambda_t = 0.01$). Units are not included in this figure, because they can be selected later. Parameter values are selected somewhat arbitrarily, for illustration purpose.	130
3.8	(a) Two tetrahedral elements with midplane and (b) corresponding cohesive zone formed by 6 nodes to which the resulting traction force is applied. . . .	130
3.9	Geometry involved in collision response. The colliding node p_o is located inside the object close to the surface triangle (p_1, p_2, p_3). The closest point on the surface is p_c . The distance between the colliding node and the surface is d	131
3.10	Stress-strain curves (top) and damage accumulation (bottom) for cylindrical sample with 500 grains loaded to failure in compression. Engineering stress at the top of the cylinder is used as a stress measure. (a) complete process showing 1365 computational steps. (b) the last 750 steps.	136
3.11	Accumulation of damaged cohesive zones in compressive test in a cylinder containing 1000 grains. Only pre-fail stages are shown, where the crack did not yet develop.	137
3.12	Propagation of crack in compressive test of a 1000-grain cylinder. Damaged cohesive zones are present in the model during the simulation, but are not shown on this figure for clarity. Time values are shown as offsets from $t = 0.638336$ s.	138
3.13	Damaged cohesive zones in tensile tests of a sample containing 1250 grains. . . .	138
3.14	Modeled fracture strength versus inverse square root of grain size for tests in (a) compression and (b) tension.	139
3.15	Comparison of the simulated compressive tests (Fig.3.14a) with the experimental data (Fig.3.3a).	139
4.1	While processing the elements in parallel, a race condition occurs when two or more elements share a node.	154
4.2	Displacements are prescribed for anchored nodes, hence their positions are known. In turn, free nodes are numbered to identify their coordinates in the resulting system of equations.	157

4.3	A cohesive zone that connects two triangular facets.	164
4.4	Traction-separation relation in the normal direction. The blue curve shows the loading/softening path; the red line shows the unloading/reloading path. $\delta_{threshold}$ denotes the failure criterion used at the implementation stage.	165
4.5	Nodes involved in collision response. A penetrating point p_1 is located below the surface of another object. The closest surface point p_c belongs to surface triangle p_2, p_3, p_4	167
4.6	Flow diagram for a single simulation step.	170
4.7	CSR matrix is defined by arrays: rows, columns and values.	172
4.8	Per-element computation consists of fetching data from global memory, computing entries of the linear system and writing the results into global memory.	174
4.9	Stages of collision detection and response: (a) detecting element-element intersection, (b) searching for the closest surface triangle among the adjacent ones, (c) computing the distance to the closest triangle and the penalty forces between the interacting nodes.	175
4.10	Setup of a uniaxial compression test.	178
4.11	Uniaxial compression of a cylinder with uncoupled unloading/reloading. Damaged cohesive zones are shown in green and fracture surfaces are shown in brown: (a) initial configuration, (b) sample showing accumulation of damage just before failure, (c) sample at failure.	180
4.12	Stress vs. time. The loading curve is linear up to the point of failure.	180
4.13	(Top) stress, (middle) reduction exponent ζ_n , (bottom) stacked plot of cohesive zones in various states vs. step number n . The reduction exponent does not return to its initial values of 0-4.	181
4.14	Uniaxial compression of a cylinder with thermodynamically consistent formulation: (a) initial configuration, (b) sample showing accumulation of damage just before failure, (c) ejection of fragments after failure.	182
4.15	(Top) stress, (middle) reduction exponent ζ_n , (bottom) stacked plot of cohesive zones in various states vs. step number n	183

4.16	Uniaxial compression test of "brittle" material with high Young's modulus. Step 890, $t = 8.08$. Damaged CZs are not shown. Initial time step is 0.05. . .	184
4.17	Uniaxial tensile test for (a) brittle; (b) ductile cohesive zones. Damaged CZs are shown in green; failed CZs are shown in brown color.	185
4.18	(Top) stress-strain curves and (bottom) time reduction exponent for brittle and ductile tensile tests.	186
4.19	Computational effort for the simulation of (a) 1,000-grain cylinder; (b) 10,000-grain cylinder.	189
5.1	Distance f is determined between the point \mathbf{p}_0 and its projection \mathbf{p}_c onto the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. Vectors \mathbf{e}_0 , \mathbf{e}_1 and \mathbf{v} are used in calculating f	200
5.2	Partitioning of the plane by the triangle domain. Different domains are distinguished by the values of barycentric coordinates of the projection of \mathbf{p}_0 onto the plane of the triangle.	210
5.3	The closest point \mathbf{p}_c may coincide with (a) one of triangle's vertices or (b) belong to one of the edges.	210
5.4	Simulation of falling and colliding grains performed with implicit FEM. . . .	214
6.1	An iceberg sighting on May 28, 2017, near Topsail Beach, Conception Bay South, NL.	221
6.2	Polyhedral grains connected with cohesive zones (CZ). The rest volume of CZs can be zero or non-zero. Grains may be deformable or rigid.	221
6.3	Normal component of the traction-separation relations as a function of normal and tangential separations Δ_n, Δ_t . Tangential component may be similar or different, depending on the selected parameters.	223
6.4	Setup of the four-point beam bending. Dimensions are shown in centimeters. . . .	225
6.5	Simulation step 193; $t = 4.2797$ s. Damaged CZs shown in green, fracture surface shown in red.	225
6.6	Simulation step 221; $t = 4.5106$ s.	225

6.7	Simulation step 256; $t = 4.5216$ s.	225
6.8	Indentation force vs time.	226
6.9	Relationship between the grain size and the flexural strength of the beam obtained from 28 simulations on beams with differing grain sizes.	227
7.1	Dimensions of the L-shaped beam, indenter application point and two crack paths, which correspond to the free end fracture (shown in red color) and the fixed end fracture (shown in blue).	237
7.2	Tetrahedral mesh for FE simulation. Cohesive zones are inserted between the highlighted elements. Indenter measuring 15×15 cm applies force to the free end of the beam.	240
7.3	(a) Resulting fracture surface. (b) Force on the indenter vs. time. (c) A typical fracture surface obtained in the field test.	241
7.4	Distribution of the first principal stress over the top surface of the beam at various time steps. The units on the legends are Pascals.	242
7.5	Accumulation of damaged cohesive zones, shown in blue color, and propagating crack, shown in red.	243
7.6	Simulated fracture at the free end of the beam.	248
7.7	Parametric study for varying length of the fixed end.	249
7.8	Parametric study for varying (a) length of the free end, (b) width of the free end.	249
7.9	Parametric tests for varying (a) width of the fixed end, (b) thickness of the sheet.	250

Nomenclature

D	Rayleigh damping matrix
E	Young's modulus; total energy
E	elasticity matrix
F	deformation gradient tensor
K	stiffness matrix
M	mass matrix
R	rotation matrix of the element
P	First Piola-Kirchhoff tensor
d	penetration distance in collision model
d_i	grain size
e	strain tensor in condensed form
\mathbf{f}_{cz}	vector of forces from the cohesive zones
\mathbf{f}_{el}	vector of elastic forces
\mathbf{f}_{ext}	vector of external forces
k	stiffness constant in collision potential
m, n	non-dimensional exponents in PPR model
u	displacement field
v_i	grain volume
\mathbf{u}_n	nodal displacement vector at time step n
α, β	shape parameters for traction-separation curves
Γ_n, Γ_t	energy constants in PPR model
Δ_n, Δ_t	normal and tangential separations

Δt	initial time step
$\dot{\epsilon}$	strain rate
ζ_i	barycentric coordinate
λ_n, λ_t	initial slope indicators in PPR model
ν	Poisson's ratio
ρ	density of the material
σ	stress tensor in condensed form
σ_0	materials constant for the starting stress for dislocation movement
σ_{max}, τ_{max}	normal and tangential cohesive strengths
σ_y	yield stress
ϕ_n, ϕ_t	normal and tangential fracture energies
Ψ	potential; energy density function
$\langle \cdot \rangle$	Macaulay bracket $\langle x \rangle = \begin{cases} 0 & x < 0, \\ x & x \geq 0. \end{cases}$

1 | Introduction and Overview

1.1 Background and motivation

In many parts of the world, ice is a part of the environment, whether it forms in freshwater basins, in the sea or on land [1]. By the force of nature, it is a part of human activities. People in the northern communities often use it to their advantages, such as for ice fishing, commuting, and building shelter. Construction of winter roads, airstrips and the support of drill rigs is also common [2]. In certain parts of Russia, it is still common to construct seasonal crossings from ice mixed with hay [3], a composite material known as Pykrete. A more exotic use of ice is progressive removal of layers from the surface of a river to form a cave to gain access underneath a ship, which is done to perform repairs of the hull and propellers [4]. While such a process is laborious and carries dangers of accidentally flooding an unfinished cave, in the Russian Far North it is a cheaper alternative to placing ships in dry docks.

Mechanical properties of ice are often studied with the ultimate goal of predicting the loading forces on ships and structures. Examples include bridges in ice-prone areas, lighthouses, oil platforms, and icebreaking ships. In each case, ice loading forces occur differently. For example, ice may be present in the form of a solid cover, rubble, icebergs or even shattered plates that look like glass [5]. A load may increase gradually or may happen instantly as a result of

a collision. Each process has random aspects, and there is no single solution to encompass all interaction scenarios.

When ice masses and structures interact in compression, the loading forces are transmitted through the high-pressure zones [6]. Instead of being distributed evenly, the pressures at the ice-structure interface are continually evolving as the ice breaks up. Crushed portions of ice are ejected from the interaction area to be replaced by the nearby intact material. Shapes and sizes of the interacting fragments are entirely random, and so are their strengths. The resulting loading forces depend on the type of ice and the interaction rate. Faster rates lead to greater stress concentrations, higher dissipation of energy and, consequently, more spalls. The forces may have an oscillating pattern, with the force-time plots typically resembling saw teeth. At low interaction rates or for certain weak types of ice, the force-time relationship may be steady.

During the interaction processes, intact solid ice is converted into powder, slush, tiny fragments, recrystallized solid, and other forms. Presence of snow is also frequent. From the perspective of chemical composition, the form is not important, but the mechanical behavior of each type of material is entirely different. And from the standpoint of computer modeling, the presence of various forms of ice creates a difficult situation. Specific modeling methods exist for solids, fluids, granular materials, rigid fragments, fracture processes, and so on. Some of them combine well; for example, liquids and rigid objects are often modeled in the same space [7] to create realistic-looking video effects. But modeling the interaction of granular materials (crushed ice), snow and intact deformable solids in the same framework proves to be a difficult task.

Computer simulations are performed to gain additional insight into a given problem. A finite element method (FEM), for example, can highlight areas of high stress concentration that are not directly observed in the experiment. Other simulations allow to analyze hypothetical scenarios that are difficult to reproduce. In Chapter 7 of this thesis, results from over 160 simulations are used to support a general formula for the strength of L-shaped beams, whereas experimental data is limited to only 9 measurements.

One motivating factor for ice mechanics research, in general, is the practical application of the obtained knowledge to the development of arctic resources, navigation in sea ice conditions and design of structures that withstand these conditions. In practice, the types of structures and vessels, the kinds of ice, and the characters of their interaction vary significantly. The behavior of ice is complex, the involved geometries are irregular, and the environmental conditions are dynamic. There is a lot of unexplored territory in the understanding of the processes of ice failure, its interaction, and distribution of the loads. In particular, redistribution of stress in a sample due to the accumulation of damage is not entirely understood.

1.2 Approaches to ice modeling

A simple method for modeling ice behavior that is often used in practice is the application of empirical equations, which are based on ample experimental data and offer accurate results quickly. A formula that describes ice thickness as a function of freezing degree days [8] is an example of a mathematical model. For many engineering tasks, simple models work quite

well.

Models are usually designed to make predictions, the more accurate, the better. Accuracy can be improved by many ways, and more complex models are not necessarily more accurate. An example of this is overfitting – "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably" [9]. However, in many cases, improved accuracy comes together with the increased complexity of the model, in particular, by considering more factors. In the case of ice thickness formula, a more sophisticated dynamic calculation would track the daily ice growth. The drawback of the improved accuracy is the laborious calculation process – a numerical integration of a differential equation. Increasing the complexity in this way is not always advantageous. Aside from improving the accuracy, a goal for developing more complex models may be an investigation of the dynamics of the process or analysis of events that cannot be directly observed.

Before a computer simulation takes place, it is formulated mathematically in the form of equations, which may be empirical formulae or may be derived from the laws of physics. What is known as “modeling methods” are usually universal numerical techniques for solving differential equations. Several families of such methods exist.

One of these techniques is the smoothed-particle hydrodynamics (SPH), where grouped particles represent continuous matter [10]. Their concentration determines the density of the material in a small surrounding volume, hence the term “smoothed.” The method can solve

any differential equations, including those for elastoplastic material deformation. This method finds its applications in many areas of physics and engineering. It carries both the benefits and the pitfalls of working with particles. On the one hand, its parallel implementation is relatively easy, and recent simulations run as many as two trillion particles [11] on supercomputers. SPH allows to combine fluids and solids and works great for high deformation rates caused by fluid flow, fragmentation, and fracture. On the other hand, the accuracy of the method depends on the number of particles in the studied volume. Runaway particles do not carry any information in the simulation, because calculated physical quantities, such as density, rely on particle interaction in groups. Boundaries of the material cannot be determined precisely, because of the inherent smoothness of particles and their constant motion. Hence, small features, such as air bubbles, would require would require very high resolution to be represented accurately. Having well-defined boundaries is essential for capturing distinct features, such as the high-pressure zones (hpzs), which are present at the interaction interface between ice and structures. Hpzs are created during spalling and crushing events, and the boundaries of the fragmented material determine their geometry. If the interaction along the borders is smoothed out, which is the case with SPH, a simulation will fail to reproduce the behavior of hpzs properly.

A technique that addresses this issue is the finite element method (FEM). Similarly to SPH, it may be applied to solve differential equations regardless of their relation to physics. Initially developed for elasticity and structural analysis, FEM quickly found its way into solving partial differential equations of many classes. It uses partitioning to find the approximate numerical solution over a given domain. For the mechanics of 3D objects, it requires partitioning the

objects into smaller elements; hence it is referred to as a mesh-based method.

Mesh-based methods have certain advantages over particle methods. For example, FEM yields higher accuracy in comparison with SPH when the computation resources are the same. Boundaries of the material can be naturally represented as mesh boundaries, allowing to resolve small inclusions like air bubbles and capturing the interactions, such as collisions between fragments [12]. On the other hand, in some cases FEM calculations require finding solutions to implicit non-linear vector equations at each simulation step. With the exception of small subclass, such operations are not implemented in parallel and have a limit on the number of unknown variables. This lowers the number of degrees of freedom in the simulation in comparison with explicit particle methods. For example, the maximum number of nodes in ABAQUS is set to 3 million by default, which is much less in comparison with particle simulations. Another limitation of mesh-based methods is that large deformations can tangle the mesh, therefore only small deformations are acceptable. Mesh entanglement is the main drawback of FEM when modeling fluids and plastic materials. Remeshing techniques can be applied to solve this issue, but they introduce several problems of their own. In comparison, particle methods allow modeling large deformations without partitioning problems.

In the field of engineering for the ice conditions, FEM is applied to evaluate the structural response of the ship's hull and the walls of offshore structures. Due to the flexibility of the method, various loading scenarios can be analyzed, including the moving loads during the ice impact [13]. Guidelines for the finite element analysis of such events are suggested by Quinton et al. [14].

Particle-in-cell (PIC) method [15] and its subclass, material point method (MPM), combine the benefits of using particles with the accuracy of meshes. MPM became well-known after its use for modeling snow in Disney’s “Frozen” [16], and is generally an excellent approach to represent materials that can be granular, plastic, rigid, or transition between different states. Being a particle method, MPM naturally models fracture processes but also benefits from the mesh-based accuracy. From the computational point of view, it requires more resources due to the processing of the particles and the background mesh.

Discrete element method (DEM) [17] describes the motion of a large number of small particles that interact by collision, friction, adhesion, and other forces. While the method is quite flexible, it is best suited for materials that consist of separate, discrete particles or blocks: sand, gravel, soil, ice rubble, or brittle solids. DEM can accurately simulate a variety of granular flows and fractures. The method is applied successfully in ice mechanics for simulating the keel gouging[18] and the keel resistance [19], as well for investigating the keel strength [20]. The method is best suited for scenarios that produce fragmented material, particularly ice rubble [21]–[23]. DEM does not treat matter as a continuum, and therefore does not rely on the equations of continuum mechanics. Instead, it solves ordinary differential equations of particle motion, and the continuum material properties emerge as the effect of large aggregate of particles. The parameters of particle interactions are adjusted to reproduce the intended behavior of the medium.

The methods mentioned above, as well as many others, allow modification, extension, improvements and fine-tuning to fit a particular scenario. In FEM, cohesive zones can be added



Figure 1.1: Pack ice event on March 30, 2017. Near St.John's, Newfoundland.

to the elastic material to account for fracture and damage. The presence of such damage alters the distribution of stress in the object, giving a different result from linearly elastic materials. Various modifications also allow adjusting the modeling methods for small or large scale microstructure, which is essential for the mechanics of ice.

In small-scale lab experiments, deformation of ice is caused by the application of indentation forces. Samples of pure ice may behave as elastic solid, and equations of elasticity and fracture strength may describe aspects of such behavior. Modeling methods, such as FEM, would be appropriate in this scenario. On the other hand, large rubble fields in nature obey a different set of rules, with their motion guided by the forces of wind and currents (Figure 1.1). Interaction between the fragments of rubble can be represented via force chains [24], which is treated better with DEM approach, as on a large scale the material is no longer solid, but granular.

The traditional ways of solving differential equations and the corresponding simulation tech-

niques are often modified to make them suitable for a particular process. For example, wing cracks may be added to a finite element formulation to model fracture [25]. Computer simulation in ice mechanics is still a developing research topic, and new techniques will continue to appear. Some researchers use the existing and well-tested software packages for modeling [26], but if the research goal itself lies in developing new and more efficient implementation, then working with source code is inevitable. Writing a custom implementation carries many difficulties and even risks, but some researchers prefer to write new implementations of the algorithms [27].

One work that utilizes a custom source code is based on discontinuous Galerkin method in combination with a specific constitutive model for ice fracture [28]. The authors optimized their algorithms to run on high-performance systems with distributed memory. Numerical results, which show fracture patterns and specimen strengths, are compared to the field experiments. The tests include uniaxial compression and beam bending – the two most common tests in ice mechanics. According to the authors, their "mechanical-mathematical model of ice adequately describes the real physical processes of ice fracturing and its design parameters are consistent with the real ones". For more information the reader is referred to their original manuscript [28].

Particle-based methods are well suited for modeling fracture or materials with high deformation, avoiding problems with mesh entanglement. The Discrete Element Method may be applied either in 2D or 3D and reproduce processes on a variety of scales, ranging from huge ice covers [29], [30] to more confined areas [31], [32] to smaller laboratory-sized specimens

[33]. The recent review by Tuhkuri [34] provides more information about discrete element modeling for ice-structure interaction.

Application of rigid-body physics for modeling ice floes is quite common. For example, Lubbad and Loset [35] utilize NVIDIA's PhysX library to produce a simulation of ice-ship interaction in real time. Their simulator is divided into several subsystems that are coupled together and communicate through network. The Mathematical Ship Model processes quantities such as power, desired speed and heading, given by the instructor. The Mathematical Ice Model calculates the interaction forces. The states of both models influence the interaction process, such as fragmenting of ice, and are displayed in the Visual System. The Ice Breaking Module is tasked with breaking up the initially intact ice cover, with the fracture pattern coming from stress distribution in semi-infinite plate resting on an elastic foundation. The formulae for the calculation of forces, the integration scheme and other details can be found in their manuscript [35].

A more recent work by Metrikin et al. [27] discusses the possibility of extending Lubbad and Loset's work [35] with the cohesive zone model (CZM) to create more realistic physically-based fractures. In another work by Dudal et al. [36], rigid body physics is applied in conjunction with computational fluid dynamics (CFD) to account for the flow of water in the area of interest.

Modeling ice-ship interaction is not a recent topic, and attempts at creating full-scale computer simulations were made several decades ago. For example, the work performed by Jebraj et

al. [37] utilizes a dynamic FEM to model interaction forces during the transient nonlinear dynamic response of collisions between ice-field and a ship. The applied finite element method used eight-node plate elements. The impact forces that cause failure in the ice-field have been determined for various parameters such as the velocity of impact, frame angle of the ship and thickness of the ice-field.

1.3 Modeling large deformation and fracture

For small deformations, pure crystalline ice at cold temperatures can be viewed as a linearly elastic material. In general, however, ice is not purely elastic, viscous or plastic. The term "viscosity" is applied to ice when referring to its permanent, non-elastic, deformation under applied load. Such deformation may be exhibited by individual ice crystals, polycrystalline ice as well as by aggregates with complex structure, i.e. rubble formations, glaciers, etc. The diversity of naturally occurring ice makes it difficult to apply classical methods of engineering analysis [38].

Even though ice can be harmful to ships and structures, it is considered to be an extremely brittle material [38]. Attempts at modeling its behavior face the problem of capturing the fracture process in one way or the other. Analytical formulations, such as linear elastic fracture mechanics [39], are applicable only to narrow cases with specific geometries and material properties. A numerical fracture modeling method is necessary to encompass a realistic scenario. One comprehensive survey of such methods is written for the application to computer

graphics [40]. In the area of graphics, a variety of approaches to fracture exist, some of which are motivated by creating an appearance of plausible patterns, rather than reproducing the physics of the process. Other, physics-based, approaches are applicable to engineering tasks. They can be grouped into the following categories: (1) mass-spring models, (2) finite element methods, and (3) meshless methods.

1.3.1 Fracture methods in computer graphics

Engineering applications usually rely on traditional methods, or even on the established software packages, to perform the analysis of fracture processes. The exception is probably modeling of hydraulic fracture, which has been in demand in recent years and shows a multitude of new tools. In computer graphics, however, researchers and artists can choose any method that gives plausible results, without much regard to its accuracy or physical relevance. In these modeling methods, the calculated forces and other physical quantities can deviate from the ones measured in the experiments, as long as the simulation results have the desired appearance. The lack of strict boundaries and requirements stimulates innovation, creation of experimental techniques, some of which may eventually find their way to engineering applications.

Koschier et al. describe one of such methods in "Adaptive Tetrahedral Meshes for Brittle Fracture Simulation" [41], which works for brittle 3D solids. It combines the finite element approach with Rankine condition [42] for the failure criterion, i.e. the material fails if the

tensile stress of a node exceeds the material strength. The essential component of Koschier’s method is the adaptive refinement of tetrahedral mesh, which occurs as the crack propagates. Another key component is collision detection and response, which handles the interaction of the newly created fragments. The method applies to various situations and has a reasonably low computational complexity. While it focuses on the plausible appearance of the fracture surface, the model is accurate in many cases of brittle fracture. It may be applied to modeling breakup of ice rubble, for example, where each piece usually has a unique 3D shape that interacts with neighboring fragments.

A failure criterion that is based on stress threshold alone has a problem with *artificial shattering* – incorrect modeling of fragmentation from the perspective of physics. While there are ways to mitigate this problem, some methods prefer to avoid it altogether by applying different fracture algorithms. The work by Hahn and Wojtan [43] is based on linear elastic fracture mechanics (LEFM) and uses the strength and toughness parameters in a physically correct way. The method uses a combination of low-resolution boundary element method (BEM) with high-resolution fracture surfaces. The method applies to granular materials, where the strength of the grains may be different from the strength of grain boundaries. Similarly to other methods in computer graphics, it focuses on the visual detail, but its accuracy and reliance on physics make it a feasible approach for modeling ice.



Figure 1.2: Fracture of thin sheets. (a) Stacking of thin plates of freshwater ice [5]. (b) ARCSim finite element model of sheet crushing against a spherical obstacle.

1.3.2 Optimization for a 2D case

Sometimes, natural ice takes the shape of large thin plates of uniform thickness (Figure 1.2a). Such an event often happens on the lakes or other freshwater reservoirs where the weather conditions are calm, and the water surface stays relatively still. From the modeling viewpoint, the fact that the ice is uniform can be used to simplify the simulations. For such material, the fractures will be perpendicular to its surface and will be caused by flexure, shear or tension. This process can be viewed as two-dimensional, significantly simplifying the necessary computations. Complex tasks, such as collision detection and remeshing, are easier to implement in 2D than in 3D and require less computational resources.

Such a model was implemented and distributed as open-source package [44], although it is missing several features needed for proper application to ice mechanics. The model was initially developed to reproduce the behavior of cloth in computer graphics [45] and subsequently

extended to reproduce plasticity [46]. It uses the tensile stress criterion to model crack propagation, which yields accurate results due to the remeshing algorithm. Depending on the selected parameters of elasticity, plasticity, and strength, the simulation captures the fracture of such materials like rubber, glass, paper and metal foil. Thin uniform ice is quite similar to glass – both exhibit brittle fracture, and ARCSim would be the right choice of software to reproduce such behavior (Figure 1.2b). Since it was not explicitly designed to model ice, ARCSim lacks buoyancy forces, typically needed to model the floating rubble. Also, it is suboptimal when handling collision response and solving sparse linear systems, as it uses a somewhat outdated solver library and an approximate method for calculating collision forces.

1.3.3 Modeling with cohesive zones

Cohesive zone model (CZM) is a method of modeling fracture where the separation of the surfaces takes place gradually. This approach can capture ductile behavior and track the slow accumulation of damage in a specimen. This feature is essential when dealing with polycrystalline materials that tend to get weaker progressively as the internal bonds get affected. In comparison with other methods, CZM has the following advantages [47]:

- The presence of an initial crack or notch is not needed; the method can predict the behavior of intact materials.
- Size of the non-linear fracture zone need not be negligible in comparison with the dimensions of the object.

A single cohesive zone is formulated mathematically in the form of traction-separation relation, i.e., the function of the closing force vs. the size of the opening of the zone. Formulations can be classified in different ways:

- Degrees of freedom for the separation. Does the formulation distinguish between normal and tangential separations?
- Whether the forces in the model are conservative (derived from a potential). Is the fracture energy the same for separation in normal and tangential directions?
- Continuity of the second derivatives of the traction forces. Will the model work with implicit integration methods?
- Are cohesive zones inserted at the beginning (intrinsic) or during the simulation (extrinsic)?

A comprehensive review of CZ formulations is done by Park et al. [48].

At temperatures significantly below the melting point, ice is usually solid and brittle, but in some circumstances, it can also exhibit flow and viscosity. For example, under sustained load, it shows creep behavior – slow plastic deformation. Crushed ice mixed with water becomes a slurry and exhibits fluid-like behavior. Particle methods are often used to model such variable behavior. However, in recent years, there have been significant advances in utilizing meshed-based approaches for modeling plasticity and fluid flow. When working with large deformations, the main issues are mesh entanglement and the need for higher resolution in

the affected areas. Wicke et al. [49] addresses these problems by adaptive local remeshing – procedure that replaces as few tetrahedra as possible, avoiding the numerical issues associated with full remeshing. Their model can handle materials with varying degree of plasticity. For purely elastic materials the advantage is the improved accuracy of the elastic force calculations in the deformed areas. For more realistic results the authors combine their method with a simple fracture algorithm and collision detection.

After solid materials break up, the newly formed fragments tend to remain separate even if they come into contact again. With ice, it is not always the case. For example, fragmented pieces of ice can form freeze bonds under external pressure or in the presence of water. Under high pressures, damaged ice can recrystallize and sinter, forming a new intact solid. Snow, powdered ice and slush mix easily upon contact, like any other fluids and granular solids. A special mesh merging algorithm is required to account for these types of behaviors in computer simulations [50]. Mesh merging allows combining elastic, plastic and liquid materials in a single framework or even within a single mesh. This approach was developed for computer graphics, but its principles apply to engineering applications and may become more popular in future.

1.4 Research method and scope of work

In the present work, a new approach has been developed and implemented, which combines and expands several modeling techniques. The existing experimental data is analyzed using

this approach. Certain assumptions are made about the material to simplify the model. The ice is treated as solid polycrystalline material, thus avoiding the issues with mesh merging and entanglement. Cohesive zone method is selected to represent fracture, allowing one to model several loading scenarios accurately. Effects of pressure softening, recrystallization and intragranular fracture are not taken into account in the present model. Within the proposed framework, intragranular fracture can be approximated by inserting additional cohesive zones within the grains.

A significant effort has been dedicated to implementing the model efficiently from scratch, which has entailed making technical choices about the use of third-party libraries, programming languages, sparse matrix format and various approaches to performing calculations. Some of the utilized algorithms are well known and freely available. They include, for example, hash tables and dictionaries, calculation of eigenvalues and eigenvectors, singular value decomposition, sparse linear solver, geometrical primitives for 2D and 3D operations, meshing library, and other procedures. Other algorithms were implemented from scratch, for example, the calculation of the Hessian and the gradient of the distance between a point and a triangle.

This thesis attempts to improve the understanding of the fracture behavior of ice by formulating and applying new models to the small- and medium-scale fracture processes observed in the lab and in the field.

1.5 Theoretical background

This section presents the mathematical background needed to understand the principles of object deformation and fracture. The aim of this introduction is to set a common basis necessary for the comprehension of subsequent chapters. For more detailed information on continuum mechanics and finite element method, several excellent books are available [51]–[53].

1.5.1 Deformation and strain

Deformation of a continuous medium can be viewed as a displacement of "points" of an object to a new location in space. This relation can be formalized as a function of the form $\mathbf{x}(\mathbf{X})$, where \mathbf{X} is the initial position of a point of the object and \mathbf{x} is its new, displaced, position. This concept applies to fluids, deformable solids and even to particle representations of matter. Only two deformation states are considered, the initial and the final, without going into the deformation history. Figure 1.3 introduces the *displacement vector field* $\mathbf{u}(\mathbf{X}) = \mathbf{x}(\mathbf{X}) - \mathbf{X}$ and illustrates what happens to the nearby points P_0 and Q_0 in a deformed object. The initial distance and the direction between the points P_0 and Q_0 are defined by the vector $d\mathbf{X}$, which, after deformation, becomes $d\mathbf{x}$. For an infinitesimally small $|d\mathbf{X}|$, the relationship between $d\mathbf{X}$ and $d\mathbf{x}$ can be linearized as

$$dx_i = \frac{\partial x_i}{\partial X_j} dX_j, \quad \text{or} \quad d\mathbf{x} = \mathbf{F}d\mathbf{X}. \quad (1.1)$$

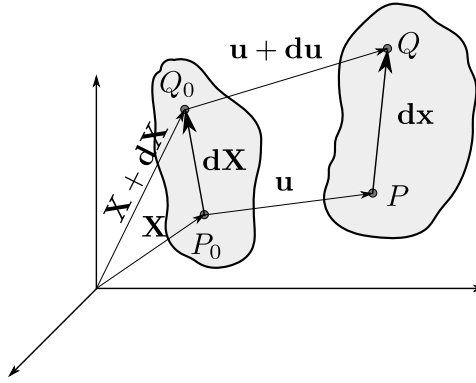


Figure 1.3: Initial and final configurations of a deformable object.

Where $\mathbf{F} = [\partial x_i / \partial X_j]$ is the *deformation gradient*. From the perspective of linear algebra, \mathbf{F} is a linear transformation that depends on the location \mathbf{X} . Note that in order for \mathbf{F} to exist, each of the derivatives $\partial x_i / \partial X_j$ must exist, i.e. deformation function must be differentiable. The deformation gradient can be defined in any pair of coordinate bases, and changing either basis transforms the values $\partial x_i / \partial X_j$ in a certain way, thus giving it an extra property, due to which \mathbf{F} is called a tensor.

Similarly, the *material displacement gradient* is defined as

$$\nabla \mathbf{u} = \left[\frac{\partial u_i}{\partial X_j} \right] = \mathbf{F} - \mathbf{I}. \quad (1.2)$$

The concepts of the deformation gradient and the displacement gradient apply to any materials, including fluids, for which deformation is usually permanent. Each new state of the fluid is the new "rest" state. Elastic solids, however, tend to return to their initial rest state once the external force is removed. When the components of the displacement gradient are small relative to unity, it is appropriate to introduce the *infinitesimal strain tensor*, which has a simple

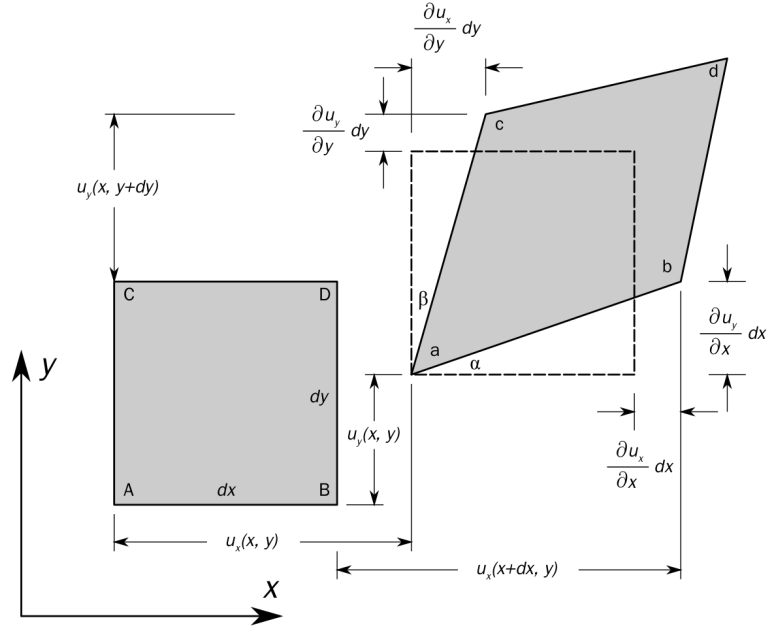


Figure 1.4: Deformation of a 2D infinitesimal element.

geometrical meaning (Figure 1.4).

The infinitesimal strain tensor is related to the displacement gradient by the following equation:

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \begin{bmatrix} e_{xx} & e_{xy} & e_{xz} \\ e_{xy} & e_{yy} & e_{yz} \\ e_{xz} & e_{yz} & e_{zz} \end{bmatrix}. \quad (1.3)$$

The strain tensor is symmetric and can be written in the condensed form (Voigt notation)

$$\mathbf{e} = [e_{xx} \quad e_{yy} \quad e_{zz} \quad 2e_{xy} \quad 2e_{yz} \quad 2e_{xz}]^T. \quad (1.4)$$

1.5.2 Strain energy

Elastic behavior is different in different materials. In metals, elasticity is due to changing distances in the lattice of atoms. For polymers, the elastic response is caused by stretching the polymer chains. Similarly, there are different ways to formulate elasticity laws. Hooke's Law for isotropic materials, for example, is formulated in terms of a relationship between stress and strain tensors:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-\nu \end{bmatrix} \begin{bmatrix} e_{xx} \\ e_{yy} \\ e_{zz} \\ 2e_{xy} \\ 2e_{yz} \\ 2e_{xz} \end{bmatrix}, \quad (1.5)$$

or, simply

$$\boldsymbol{\sigma} = \mathbf{E} \cdot \mathbf{e}. \quad (1.6)$$

Where $\boldsymbol{\sigma}$ is the vector of stress tensor components, E is the modulus of elasticity, ν is Poisson's ratio, and \mathbf{E} is elasticity matrix. Another way to describe the elastic behavior is by introducing the *energy density* function [54] per unit of undeformed volume $\psi[\mathbf{x}; \mathbf{X}]$, where \mathbf{x} denotes the

deformation function. Total energy for the deforming body can be obtained by integrating the energy density

$$E[\mathbf{x}] = \int_{\Omega} \psi[\mathbf{x}; \mathbf{X}] d\mathbf{X}. \quad (1.7)$$

The internal traction can then be obtained as

$$\boldsymbol{\tau}(\mathbf{X}) = -\mathbf{P} \cdot \mathbf{N}, \quad (1.8)$$

where \mathbf{N} is the outward unit normal to the boundary in the undeformed configuration; \mathbf{P} is the *First Piola-Kirchhoff stress tensor*. For materials where $\psi[\mathbf{x}; \mathbf{X}]$ is defined (hyperelastic materials), First Piola-Kirchhoff tensor is

$$\mathbf{P}(\mathbf{F}) = \partial \psi(\mathbf{F}) / \partial \mathbf{F}. \quad (1.9)$$

Note that this requires an explicit expression of ψ as a function of \mathbf{F} .

This approach can be extended to discretized formulations, e.g. in finite element method, as described by Sifakis [54]. Obtaining forces acting on the nodes of finite elements is easy, and the resulting formulations work for large displacements, and even for degenerate or inverted elements (finite elements whose volume was inverted due to deformation). In the presented work, however, a different approach was used, which is slightly computationally efficient, but works within the limitation of small displacements.

1.5.3 Cauchy momentum equation

The Cauchy momentum equation is a partial differential equation (PDE) in a vector form that describes the motion of the continuous medium. Based on Newton's second law it states that the time rate of change of momentum in an arbitrary portion of the continuum is equal to the resultant force acting on the considered portion. It is written (in the Lagrangian form) as

$$\frac{D\mathbf{v}}{Dt} = \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} + \mathbf{g}, \quad (1.10)$$

where ρ is the density of the material, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, \mathbf{g} includes all body forces, \mathbf{v} is the velocity vector field and $\frac{D\mathbf{v}}{Dt}$ denotes the material derivative. This PDE is the basis of the proposed simulation and is solved numerically using FEM. Discretization is applied to the object by partitioning it into a finite number of elements. The time component is also discretized by selecting time steps.

1.5.4 Finite element method

The preceding sections describe the laws of physics that address the behavior of elastic materials as continuous mass. The laws are formulated in the form of partial differential equations, with analytical solutions available for a limited number of cases. To address the problem in general, a numerical approximation is obtained using the finite element method. A deformable body is represented as a finite number of points corresponding to the nodes of a mesh. In the

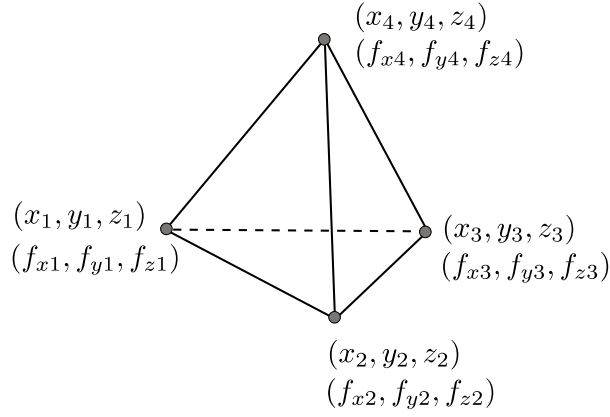


Figure 1.5: A single tetrahedral element.

simplest case, the mesh consists of tetrahedral elements (Figure 1.5), which partition the object. Deformation of the tetrahedra create elasticity forces on the adjacent nodes, which can be concisely expressed as $\mathbf{f} = \mathbf{f}(\mathbf{X}, \mathbf{u})$, where \mathbf{f} is a 12-vector of forces on the nodes of the element, \mathbf{X} is the vector of the initial positions of the nodes, and $\mathbf{u} = \mathbf{x} - \mathbf{X}$ is the difference between the initial and current nodal positions, i.e., the vector of nodal displacements.

The force function $\mathbf{f}(\mathbf{X}, \mathbf{u})$ can be formulated in several ways. The most trivial way is by representing the edges of the tetrahedron as elastic springs – a formulation that does correspond to elastic material with unnatural properties. For more realistic results, the behavior of the solid material (that fills the tetrahedron) must be considered. Elastic forces on the element's nodes can be obtained by differentiating the strain energy of function of the element:

$$\mathbf{f} = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}. \quad (1.11)$$

This method works for the cases when the constitutive model of the material is defined by a strain energy density function and is described in detail by Sifakis et al. [53]. Such constitutive

models are called hyperelastic. There are several common formulations of the energy density function, which correspond to different elastic response for large deformations, hence modeling different natural materials. For small deformations these formulations are nearly identical, yielding linear elastic response. The derivations are straightforward, and the resulting expressions for numerical computation are quite simple.

In some natural materials, deformations are inherently small. For example, rocks and concrete will shatter before deformations even become visible to the naked eye. For some types of ice, this is also the case. If the relationship between the deformation and the stress is assumed to be linear, then the nodal forces $\mathbf{f}(\mathbf{X}, \mathbf{u})$ can be expressed as $\mathbf{f} = \mathbf{K} \cdot \mathbf{u}$, where \mathbf{K} is the element stiffness matrix.

To proceed with the derivation, we must establish how the values of nodal displacement interpolate inside the tetrahedron. For linear tetrahedron this is done by introducing the tetrahedral natural (barycentric) coordinates $(\zeta_1, \zeta_2, \zeta_3, \zeta_4)$. The transformation between the arbitrary barycentric coordinates and the corresponding Cartesian coordinates (x, y, z) is given by

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix}, \quad (1.12)$$

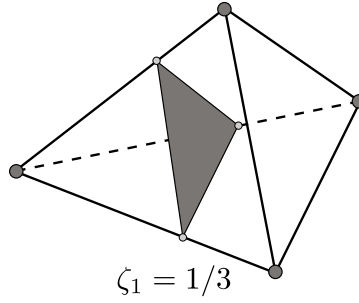


Figure 1.6: A plane defined by $\zeta_1 = 1/3$.

or, simply,

$$\mathbf{p} = \mathbf{J} \cdot \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \zeta_4 \end{bmatrix}^T. \quad (1.13)$$

The coordinates of the tetrahedron's vertices denoted by (x_i, y_i, z_i) are shown in Figure 1.5.

An equation of the form $\zeta_i = \text{const}$ defines a plane parallel to the i -th facet of the tetrahedron (Figure 1.6).

For non-degenerate tetrahedra, the mapping is invertible:

$$\begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \zeta_4 \end{bmatrix}^T = \mathbf{J}^{-1} \cdot \begin{bmatrix} 1 & x & y & z \end{bmatrix}^T. \quad (1.14)$$

The inverse of \mathbf{J} can be constructed explicitly:

$$\mathbf{J}^{-1} = \frac{1}{6V} \begin{bmatrix} 6V_{01} & y_{42}z_{32} - y_{32}z_{42} & x_{32}z_{42} - x_{42}z_{32} & x_{42}y_{32} - x_{32}y_{42} \\ 6V_{02} & y_{31}z_{43} - y_{34}z_{13} & x_{43}z_{31} - x_{13}z_{34} & x_{31}y_{43} - x_{34}y_{13} \\ 6V_{03} & y_{24}z_{14} - y_{14}z_{24} & x_{14}z_{24} - x_{24}z_{14} & x_{24}y_{14} - x_{14}y_{24} \\ 6V_{04} & y_{13}z_{21} - y_{12}z_{31} & x_{21}z_{13} - x_{31}z_{12} & x_{13}y_{21} - x_{12}y_{31} \end{bmatrix}, \quad (1.15)$$

where

$$x_{ij} = x_i - x_j, \quad y_{ij} = y_i - y_j, \quad z_{ij} = z_i - z_j,$$

$$V = \det(\mathbf{J})/6,$$

$$6V_{01} = x_2(y_3z_4 - y_4z_3) + x_3(y_4z_2 - y_2z_4) + x_4(y_2z_3 - y_3z_2), \quad (1.16)$$

$$6V_{02} = x_1(y_4z_3 - y_3z_4) + x_3(y_1z_4 - y_4z_1) + x_4(y_3z_1 - y_1z_3),$$

$$6V_{03} = x_1(y_2z_4 - y_4z_2) + x_2(y_4z_1 - y_1z_4) + x_4(y_1z_2 - y_2z_1),$$

$$6V_{04} = x_1(y_3z_2 - y_2z_3) + x_2(y_1z_3 - y_3z_1) + x_3(y_2z_1 - y_1z_2),$$

By introducing the new notation, the entries of \mathbf{J}^{-1} become

$$\mathbf{J}^{-1} = \frac{1}{6V} \begin{bmatrix} 6V_{01} & a_1 & b_1 & c_1 \\ 6V_{02} & a_2 & b_2 & c_2 \\ 6V_{03} & a_3 & b_3 & c_3 \\ 6V_{04} & a_4 & b_4 & c_4 \end{bmatrix}. \quad (1.17)$$

It can be noted that

$$\frac{\partial \zeta_i}{\partial x} = \frac{a_i}{6V}, \quad \frac{\partial \zeta_i}{\partial y} = \frac{b_i}{6V}, \quad \frac{\partial \zeta_i}{\partial z} = \frac{c_i}{6V}. \quad (1.18)$$

Then, for any function $F(\zeta_1, \zeta_2, \zeta_3, \zeta_4)$, the partial derivatives can be computed as (Einstein

summation convention used)

$$\begin{aligned}
\frac{\partial F}{\partial x} &= \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial x} = \frac{1}{6V} \left(\frac{\partial F}{\partial \zeta_1} a_1 + \frac{\partial F}{\partial \zeta_2} a_2 + \frac{\partial F}{\partial \zeta_3} a_3 + \frac{\partial F}{\partial \zeta_4} a_4 \right) = \frac{a_i}{6V} \frac{\partial F}{\partial \zeta_i}, \\
\frac{\partial F}{\partial y} &= \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial y} = \frac{1}{6V} \left(\frac{\partial F}{\partial \zeta_1} b_1 + \frac{\partial F}{\partial \zeta_2} b_2 + \frac{\partial F}{\partial \zeta_3} b_3 + \frac{\partial F}{\partial \zeta_4} b_4 \right) = \frac{b_i}{6V} \frac{\partial F}{\partial \zeta_i}, \\
\frac{\partial F}{\partial z} &= \frac{\partial F}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial z} = \frac{1}{6V} \left(\frac{\partial F}{\partial \zeta_1} c_1 + \frac{\partial F}{\partial \zeta_2} c_2 + \frac{\partial F}{\partial \zeta_3} c_3 + \frac{\partial F}{\partial \zeta_4} c_4 \right) = \frac{c_i}{6V} \frac{\partial F}{\partial \zeta_i}.
\end{aligned} \tag{1.19}$$

The above derivations allow one to obtain the strain tensor. First, let's denote the nodal displacements by $u_{x1}, u_{y1}, \dots, u_{z4}$. Displacement in the interior of the element can be linearly interpolated as

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_{x1} & u_{x2} & u_{x3} & u_{x4} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} \\ u_{z1} & u_{z2} & u_{z3} & u_{z4} \end{bmatrix} \cdot \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix}. \tag{1.20}$$

Nodal displacements can be written concisely as a single vector

$$\mathbf{u}_e = [u_{x1} \ u_{y1} \ u_{z1} \ u_{x2} \ u_{y2} \ u_{z2} \ u_{x3} \ u_{y3} \ u_{z3} \ u_{x4} \ u_{y4} \ u_{z4}]^T. \tag{1.21}$$

For a given deformation of a tetrahedral element \mathbf{u}_e , it is possible to obtain the strain tensor \mathbf{e}

by combining equations (1.3) and (1.19):

$$\mathbf{e} = \mathbf{B} \cdot \mathbf{u}_e, \quad (1.22)$$

where

$$\mathbf{B} = \frac{1}{6V} \begin{bmatrix} a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 & 0 & 0 & a_4 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 \\ b_1 & a_1 & 0 & b_2 & a_2 & 0 & b_3 & a_3 & 0 & b_4 & a_4 & 0 \\ 0 & c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 \\ c_1 & 0 & a_1 & c_2 & 0 & a_2 & c_3 & 0 & a_3 & c_4 & 0 & a_4 \end{bmatrix}. \quad (1.23)$$

Similarly to expression (1.6), stress tensor in the element is expressed as

$$\boldsymbol{\sigma} = \mathbf{E} \cdot \mathbf{e}. \quad (1.24)$$

Strain energy of the tetrahedral element is given by

$$E(\mathbf{u}_e) = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma}^T \cdot \mathbf{e} d\Omega = \frac{1}{2} \int_{\Omega} \mathbf{e}^T \cdot \mathbf{E}^T \cdot \mathbf{e} d\Omega = \frac{1}{2} \int_{\Omega} \mathbf{u}_e^T \cdot \mathbf{B}^T \cdot \mathbf{E} \cdot \mathbf{B} \cdot \mathbf{u}_e d\Omega. \quad (1.25)$$

Gradient of the expression (1.25) with respect to nodal coordinates yields nodal forces. It can

be shown that

$$\mathbf{f}_e = \nabla E = \int_{\Omega} \mathbf{B}^T \cdot \mathbf{E} \cdot \mathbf{B} d\Omega \cdot \mathbf{u}_e = V \mathbf{B}^T \cdot \mathbf{E} \cdot \mathbf{B} \cdot \mathbf{u}_e = V \mathbf{K} \cdot \mathbf{u}_e, \quad (1.26)$$

where \mathbf{K} is the *stiffness matrix*.

The above derivations show a method to obtain the elastic forces of the tetrahedral element. The assumption of small displacements implies that the result is not rotationally invariant, i.e., large rotations will yield incorrect results. One easy fix is to exclude the rotational components from the displacements before evaluating forces. For the mathematical formulation of this transformation, refer to section 3.3.3.

1.5.5 Cohesive zones and grains within FEM

The FE procedure described above can be applied to a continuous deformable material to simulate the effects of elasticity. But what if the material is granular or polycrystalline? In such a case, the grains interact with one another via grain boundaries, which is reflected in equation (1.10) as boundary conditions. If the grains are not physically connected as in dry sand or gravel, then only contact/friction forces exist between grains. In other materials, such as in wet sand or polycrystalline solids, additional cohesion between grains must be included in the model.

Cohesive zone model (CZM) is a method for modeling fracture, but in addition to handling

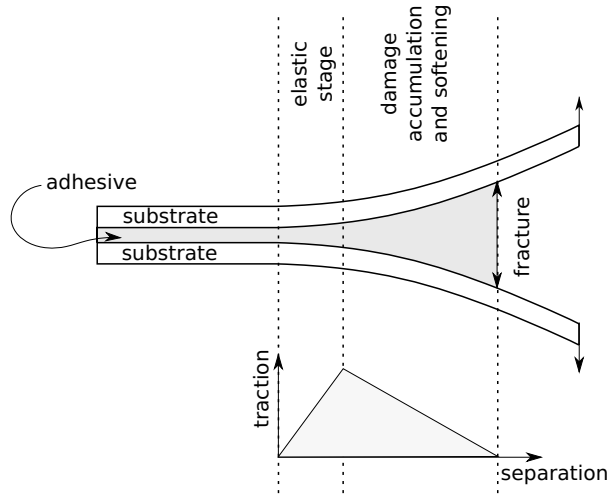


Figure 1.7: Schematic representation of a cohesive zone.

brittle solids, it can also model peeling of elastic tape, tearing of fibers in polymers and other types of debonding processes [55]. Debonding is treated as gradual, with several distinct stages (Figure 1.7). Aside from the traction-separation relationship, a formulation of CZM must select whether CZs have zero or non-zero rest volume, whether the time integration method will be implicit or explicit and whether the CZs will be inserted at the beginning or during the simulation. For application to a specific process, a model must be fine-tuned to match the rest of the framework.

1.5.6 Collision detection and response

When addressing the interaction of loose grains, the main problem is detecting the collisions between the grains, more precisely, detecting the intersections between the polygons that constitute the grains. Several varieties of collision detection algorithms exist [56], and the research work in this area is still ongoing. A naive approach is to test each colliding object against all

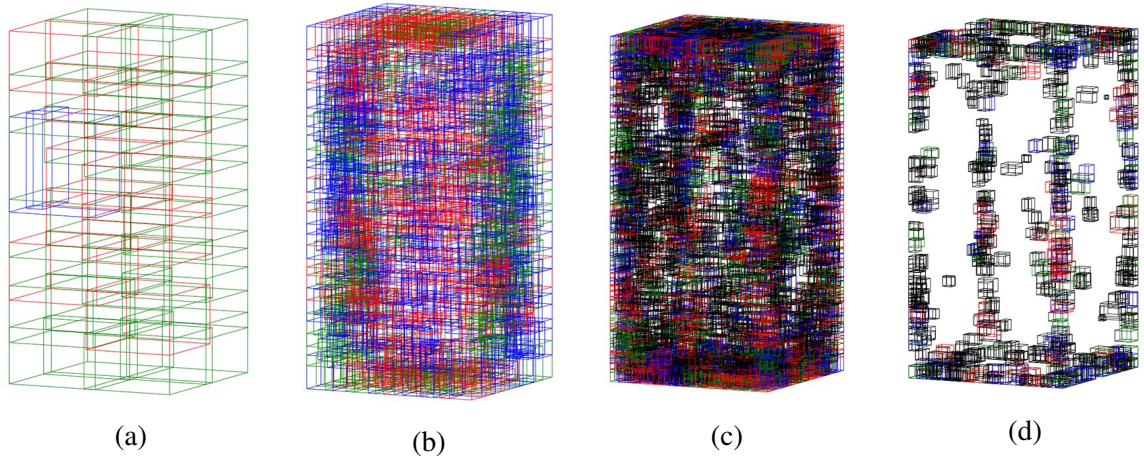


Figure 1.8: Levels of the bounding volume hierarchy going from (a) the top of the tree to (d) the leaves of the tree.

other objects, resulting in $O(N^2)$ computational complexity. To lower the computational complexity, a two-step process is employed, usually called the broad phase and the narrow phase.

The broad phase quickly eliminates a large portion of the N^2 tests by checking intersections of the bounding volumes of object groups. It is a family of algorithms based on constructing the bounding volume hierarchy (BVH). Bounding volumes (BV) are usually simple geometrical shapes, such as spheres or parallelepipeds, whose intersection can be checked by inexpensive calculations. BVs are constructed for all objects and for groups of objects, forming a BV hierarchy (Figure 1.8). Intersection checks are performed starting from the top of the hierarchy, thus eliminating large numbers of pairwise checks. One implementation of this algorithm is described by Tang et al. [57], and the source code is freely distributed.

Since the BVs are bigger than the entities they enclose, it is possible for BVs to intersect when the actual objects do not. The narrow phase checks for the intersections between the actual entities. This procedure is usually more computationally expensive but can be performed

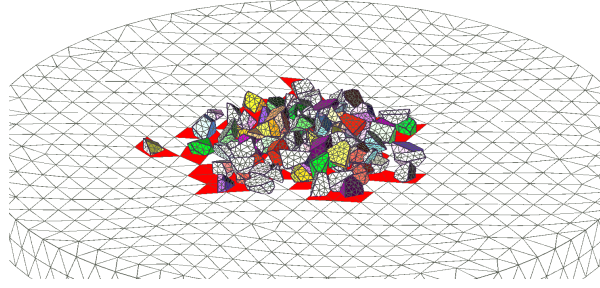


Figure 1.9: Grains rest on the top of a solid disk. Areas of contact are shown in red color.

in parallel. Smallest entities can be, for example, grains of the material, or the tetrahedral elements. For improved efficiency, it helps to identify the smallest type of object which cannot intersect itself, such as a grain.

After the intersecting entities are detected, a collision response algorithm is invoked. A simple approach is to apply penalty forces to the contacts, pushing apart the colliding objects (Figure 1.9). A completely different approach to handling collisions is by constraining the motion of the objects [58], rather than by applying forces.

1.6 Organization of thesis

This thesis is written in manuscript format and each chapter contains a review of publications that are relevant to that chapter.

References

- [1] T. Sanderson, “Ice mechanics and risks to offshore structures,” 1988.
- [2] R. M. W. Frederking and L. W. Gold, “The bearing capacity of ice covers under static loads,” *Canadian Journal of Civil Engineering*, vol. 3, no. 2, pp. 288–293, 1976.
- [3] N. N. Novgoroda. (2019). A crossing over the volga from sennaya square is being built from hay., Youtube, [Online]. Available: <https://www.youtube.com/watch?v=I3goBB3UJ28>.
- [4] C. Time. (2017). Lena: A complete vymorozka | unknown russia, Youtube, [Online]. Available: <https://www.youtube.com/watch?v=ahIigZeJDPs>.
- [5] RadiantSpiritGallery. (2016). Lake superior ice stacking (feb. 13, 2016), Youtube, [Online]. Available: <https://www.youtube.com/watch?v=HqTEac8XFQo>.
- [6] I. J. Jordaan, “Mechanics of ice–structure interaction,” *Engineering Fracture Mechanics*, vol. 68, no. 17-18, pp. 1923–1960, 2001.
- [7] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, “Unified particle physics for real-time applications,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 153, 2014.
- [8] “Petroleum and natural gas industries – Arctic offshore structures,” International Organization for Standardization, Geneva, CH, Standard ISO 19906:2010(en), Dec. 2010.

- [9] (). Overfitting, [Online]. Available: <https://www.lexico.com/en/definition/overfitting> (visited on 07/16/2019).
- [10] R. Gingold and J. Monaghan, “Smoothed particle hydrodynamics: Theory and application to non-spherical stars,” en, *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. 3, pp. 375–389, Dec. 1977, ISSN: 0035-8711, 1365-2966. DOI: 10.1093/mnras/181.3.375.
- [11] D. Potter, J. Stadel, and R. Teyssier, “Pkdgrav3: Beyond trillion particle cosmological simulations for the next era of galaxy surveys,” *Computational Astrophysics and Cosmology*, vol. 4, no. 1, p. 2, May 2017, ISSN: 2197-7909. DOI: 10.1186/s40668-017-0021-1. [Online]. Available: <https://doi.org/10.1186/s40668-017-0021-1>.
- [12] M. Müller, N. Chentanez, T.-Y. Kim, and M. Macklin, “Air meshes for robust collision handling,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 133, 2015.
- [13] B. Quinton, C. Daley, and R. Gagnon, “Effect of Moving Ice Loads on the Plastic Capacity of a Ship’s Structure,” in *ICETECH 2010*, Anchorage, Alaska, Sep. 2010.
- [14] B. W. Quinton, C. G. Daley, R. E. Gagnon, and D. B. Colbourne, “Guidelines for the nonlinear finite element analysis of hull response to moving loads on ships and offshore structures,” *Ships and Offshore Structures*, vol. 12, no. sup1, S109–S114, 2017.
- [15] F. J. Vesely, *Computational Physics: an Introduction*, English. Boston, MA: Springer US : Imprint : Springer, 2001, OCLC: 840283701, ISBN: 978-1-4615-1329-2.

- [16] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, “A material point method for snow simulation,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 102, 2013.
- [17] A. Munjiza, *The combined finite-discrete element method*. Hoboken, NJ: Wiley, 2004, ISBN: 978-0-470-84199-0.
- [18] E. B. Dudley, L. Liu, R. Sarracino, and R. Taylor, “Using discrete element model to simulate keel-gouging: A sensitivity analysis,” in *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers, 2017, V008T07A022–V008T07A022.
- [19] H. Gong, A. Polojärvi, J. Tuhkuri, *et al.*, “Preliminary 3d dem simulations on ridge keel resistance on ships,” POAC, 2017.
- [20] A. Polojärvi and J. Tuhkuri, “3d discrete numerical modelling of ridge keel punch through tests,” *Cold Regions Science and Technology*, vol. 56, no. 1, pp. 18–29, 2009.
- [21] J. Paavilainen, J. Tuhkuri, and A. Polojärvi, “2d numerical simulations of ice rubble formation process against an inclined structure,” *Cold Regions Science and Technology*, vol. 68, no. 1-2, pp. 20–34, 2011.
- [22] A. Polojärvi, J. Tuhkuri, and A. Pustogvar, “Dem simulations of direct shear box experiments of ice rubble: Force chains and peak loads,” *Cold Regions Science and Technology*, vol. 116, pp. 12–23, 2015.

- [23] V.-P. Lilja, A. Polojärvi, J. Tuhkuri, and J. Paavilainen, “Effective material properties of a finite element-discrete element model of an ice sheet,” *Computers & Structures*, vol. 224, p. 106 107, 2019.
- [24] M. van den Berg and R. Lubbad, “The application of a non-smooth discrete element method in ice rubble modelling,” in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2015.
- [25] K. Kolari, “Simulation of the temperature and grain size dependent uniaxial compressive strength using 3d wing crack model,” in *23rd International Conference on Port and Ocean Engineering under Arctic Conditions, POAC 2015*, 2015.
- [26] W. Lu, R. Lubbad, S. Løset, and K. Høyland, “Cohesive zone method based simulations of ice wedge bending: A comparative study of element erosion, cem, dem and xfem,” in *The 21st IAHR International Symposium on Ice*, 2012, pp. 920–938.
- [27] I. Metrikin, W. Lu, R. Lubbad, S. Løset, and M. Kashafutdinov, “Numerical simulation of a floater in a broken-ice field: Part i—model description,” in *ASME 2012 31st International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers, 2012, pp. 579–588.
- [28] V. Miryaha, A. Sannikov, V. Biryukov, and I. Petrov, “Discontinuous Galerkin Method for Investigating Ice Strength,” *Mathematical Models and Computer Simulations*, vol. 10, no. 5, pp. 609–615, Sep. 2018, ISSN: 2070-0482, 2070-0490. DOI: 10.1134/S2070048218050083.

- [29] A. Herman, “Discrete-Element bonded-particle Sea Ice model DESIgn, version 1.3a – model description and implementation,” *Geoscientific Model Development*, vol. 9, no. 3, pp. 1219–1241, Apr. 2016, ISSN: 1991-9603. DOI: 10.5194/gmd-9-1219-2016.
- [30] J. Åström, T. Riikilä, T. Tallinen, T. Zwinger, D. Benn, J. C. Moore, and J. Timonen, “A particle based simulation model for glacier dynamics,” *Cryosphere*, vol. 7, 2013.
- [31] D. Morgan, R. Sarracino, R. McKenna, and J. W. Thijssen, “Simulations of ice rubbing against conical structures using 3d dem,” in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2015.
- [32] R. Yulmetov, R. Lubbad, and S. Løset, “Planar multi-body model of iceberg free drift and towing in broken ice,” *Cold Regions Science and Technology*, vol. 121, pp. 154–166, Jan. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.08.011.
- [33] S. Mohammadafzali, R. Sarracino, R. Taylor, C. Stanbridge, A. Marchenko, *et al.*, “Investigation and 3d discrete element modeling of fracture of sea ice beams,” in *Arctic technology conference*, Offshore Technology Conference, 2016.
- [34] J. Tuhkuri and A. Polojärvi, “A review of discrete element simulation of ice–structure interaction,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2129, p. 20170335, Sep. 2018, ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2017.0335. (visited on 10/26/2018).
- [35] R. Lubbad and S. Løset, “A numerical model for real-time simulation of ship–ice interaction,” *Cold Regions Science and Technology*, vol. 65, no. 2, pp. 111–127, 2011.

- [36] A. Dudal, C. Septseault, P.-A. Beal, S. Le Yaouang, and B. Roberts, “A new arctic platform design tool for simulating ice-structure interaction,” in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2015.
- [37] C. Jebaraj, A. Swamidas, L. Shih, and K. Munaswamy, “Finite element analysis of ship/ice interaction,” *Computers & Structures*, vol. 43, no. 2, pp. 205–221, Apr. 1992, ISSN: 00457949. DOI: 10.1016/0045-7949(92)90138-P.
- [38] T. Sanderson, *Ice mechanics: risks to offshore structures*. London, UK ; Boston: Graham & Trotman, 1988, ISBN: 978-0-86010-785-9.
- [39] K. Abdel-Tawab and G. J. Rodin, “On the relevance of linear elastic fracture mechanics to ice,” *International Journal of Fracture*, vol. 62, no. 2, pp. 171–181, 1993.
- [40] L. Muguercia, C. Bosch, and G. Patow, “Fracture modeling in computer graphics,” *Computers & Graphics*, vol. 45, pp. 86–100, 2014.
- [41] D. Koschier, S. Lipponer, and J. Bender, “Adaptive tetrahedral meshes for brittle fracture simulation,” in *Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation*, Eurographics Association, 2014, pp. 57–66.
- [42] J. Collins, “6.2 Maximum normal stress theory (Rankine’s theory),” in *Failure of materials in mechanical design: analysis, prediction, prevention*, 2nd ed, New York: Wiley, 1993, pp. 142–143, ISBN: 978-0-471-55891-0.
- [43] D. Hahn and C. Wojtan, “High-resolution brittle fracture simulation with boundary elements,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 151, 2015.

- [44] T. Pfaff, R. Narain, J. M. De Joya, and J. F. O’Brien, “Adaptive tearing and cracking of thin sheets,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 110, 2014.
- [45] R. Narain, A. Samii, and J. F. O’Brien, “Adaptive anisotropic remeshing for cloth simulation,” *ACM transactions on graphics (TOG)*, vol. 31, no. 6, p. 152, 2012.
- [46] R. Narain, T. Pfaff, and J. F. O’Brien, “Folding and crumpling adaptive sheets,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 51, 2013.
- [47] Z. P. Bazant and J. Planas, *Fracture and size effect in concrete and other quasibrittle materials*. CRC press, 1997, vol. 16.
- [48] K. Park and G. H. Paulino, “Cohesive zone models: A critical review of traction-separation relationships across fracture surfaces,” *Applied Mechanics Reviews*, vol. 64, no. 6, p. 060 802, 2011.
- [49] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O’Brien, “Dynamic local remeshing for elastoplastic simulation,” in *ACM Transactions on graphics (TOG)*, ACM, vol. 29, 2010, p. 49.
- [50] P. Clausen, M. Wicke, J. R. Shewchuk, and J. F. O’Brien, “Simulating liquids and solid-liquid interactions with lagrangian meshes,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 2, p. 17, 2013.
- [51] G. E. Mase, *Theory and problems of continuum mechanics: Schaum’s Outline Series*. McGraw-Hill, 1970.
- [52] B. McGinty. (2012). Continuum mechanics, [Online]. Available: <https://www.continuummechanics.org/> (visited on 04/16/2019).

- [53] E. Sifakis and J. Barbič, “Finite element method simulation of 3d deformable solids,” *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, vol. 1, no. 1, pp. 1–69, 2015.
- [54] E. Sifakis, *Finite element simulation of 3D deformable solids*. San Rafael, California: Morgan & Claypool Publishers, 2016, OCLC: 889647388, ISBN: 978-1-62705-442-3.
- [55] M. Paggi and J. Reinoso, “An anisotropic large displacement cohesive zone model for fibrillar and crazing interfaces,” *International Journal of Solids and Structures*, vol. 69, pp. 106–120, 2015.
- [56] C. Ericson, *Real-time collision detection*, ser. Morgan Kaufmann series in interactive 3D technology. Amsterdam ; Boston: Elsevier, 2005, ISBN: 978-1-55860-732-3.
- [57] M. Tang, D. Manocha, and R. Tong, “Mccd: Multi-core collision detection between deformable models using front-based decomposition,” *Graphical Models*, vol. 72, no. 2, pp. 7–23, 2010, ISSN: 1524-0703. DOI: DOI:10.1016/j.gmod.2010.01.001.
- [58] D. Baraff, “An introduction to physically based modeling: Rigid body simulation ii—nonpenetration constraints,” *SIGGRAPH course notes*, pp. D31–D68, 1997.

2 | Review of Ice Mechanics

2.1 Crystal structure, microstructure, types of ice and formation

2.1.1 Crystal structure and microstructure

Mechanical properties of ice are, to a great extent, determined by its crystalline structure. At temperatures and pressures usually encountered on the surface of Earth, ice exists in the form denoted as ice Ih, where h stands for hexagonal. Eleven other crystalline forms exist [1], which are not stable at normal temperatures and pressures. In the lattice of ice Ih, every molecule is bonded to four other molecules by hydrogen bonds, forming the layers of hexagonal "sheets." In each layer of the basal plane the angle between adjacent oxygen atoms is approximately 109.5° , whereas the atoms H-O-H in the water molecule form an angle of approximately 104.5° . The resulting crystal geometry is shown on Figure 2.1. Bonding between layers is more sparse than within layers, giving the crystal anisotropic properties. The direction perpendicular to the basal plane is denoted as c-axis. The orientation of hydrogen atoms is random [2], but two (Bernal-Fowler) rules must be fulfilled: (1) there are two protons near each oxygen; (2) only one proton lies on each hydrogen bond between two oxygen atoms. Deviation from these rules creates Bjerrum defects (Bjerrum called them orientation defects),

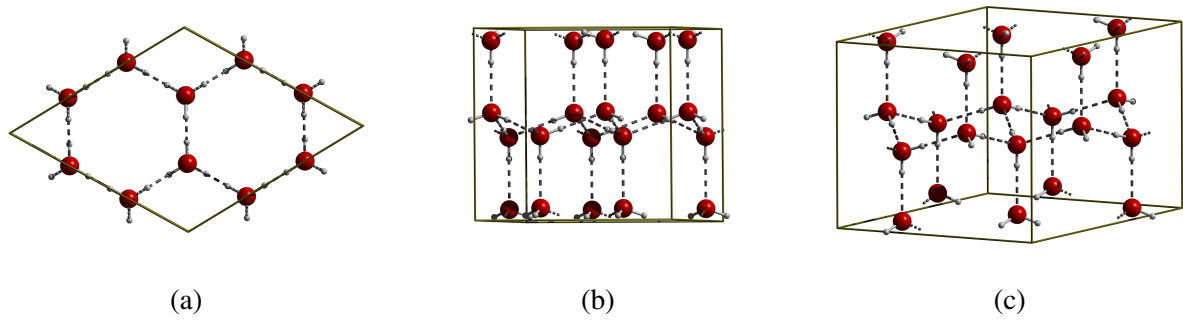


Figure 2.1: Crystal structure of the ice Ih (a) along the c-axis, (b) perpendicular to c-axis, and (c) at a free angle. Oxygen atoms are shown in red; hydrogen atoms are shown in grey. Hydrogen bonds are shown as dashed lines. The arrangement of hydrogen atoms is random; one possible arrangement is shown.

which either leaves a hydroxide ion OH^- or a hydronium ion H_3O^+ in the lattice.

The crystal structure is important in describing creep and fracture because the movement of dislocations through the crystal explains certain types of deformation. Ice crystals are of extremely high purity [3], regardless of the initial concentration of solutes in the water, as the impurities are rejected during freezing. For example, solubility of HCl is only 3×10^{-6} at -10°C . Other ions and compounds, such as, F^- , HF, NH_4^+ , NH_3 , NH_4OH , NH_4F , can incorporate into the crystals at higher concentrations. The Bjerrum defects are the presence of hydronium and hydroxide ions along with either two or zero protons per bond. A bond with two protons is denoted as D-defect, and a bond with zero protons is an L-defect. Since the orientation of hydrogen atoms is random, a passage of dislocation is expected to produce a disorder among the hydrogen atoms and cause breaches of Bernal-Fowler rules (creating new hydroxide and hydronium ions) [4].

Plastic deformation in ice crystals occurs mainly by slip on the basal plane – dislocations that

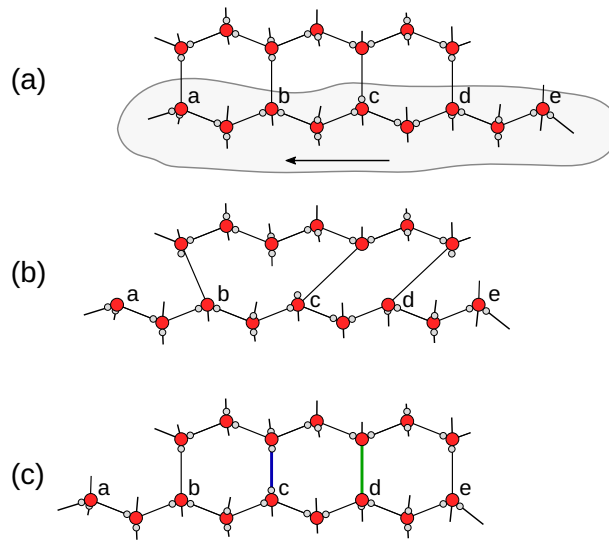


Figure 2.2: Movement of dislocation through the ice crystal. After Glen [4]. (a) Initial hydrogen configuration follows Bernal-Fowler rules, with one hydrogen per bond. (b) The bottom layer of molecules slides to the left, allowing the passage of dislocation. (c) The new configuration does not follow Bernal-Fowler rules. One bond site now contains two hydrogen atoms, highlighted in blue. Another bond site contains no hydrogen atoms, highlighted in green.

were observed directly by Hayes and Webb [5]. After a dislocation passes, the arrangement of oxygen atoms remains the same, but each oxygen at the top layer changes partners with the one at the bottom layer. Since the orientation of hydrogen atoms is random, the overall arrangement changes to a new state (Figure 2.2). Using the calculations based on energy, Glenn [4] estimated the stress needed to deform an ice crystal. The calculated value turned out to be a few orders of magnitude higher than the observed strength. Glenn concluded that the defects move through the lattice continuously, making it possible for the dislocation to travel at lower applied stresses. Schulson [2] mentions that precise way in which dislocations overcome the energy barrier is unknown.

Only one crystalline form of ice exists in normal conditions, ice Ih, but its microstructure may be different, depending on the growth conditions and the solutes present in the water

before freezing. According to Owston and Lonsdale [6], almost all ice in nature is crystalline, whereas monocrystalline ice exists primarily in hailstones, which are composed of one or several individual crystals.

After nucleation, the crystals can grow in any direction, and the growth rate depends on the temperature of the surrounding water. In the basal plane, the crystals grow faster than along the c-axis [7]. When the growth in the basal plane is constrained, the crystal continues to grow along the prismatic plane (c-axis). The relatively sparse hexagonal arrangement of molecules in the crystal results in the lower density (916.7 kg/m^3 at 0°C) of ice than that of water (999.83 kg/m^3 at 0°C). A notable property of freshwater is that its density increases from 999.83 kg/m^3 at 0°C to 999.96 kg/m^3 at 4°C , contrary to what would be expected due to thermal expansion. One probable explanation is that to some extent the crystal structure of solid ice Ih is still present within this range of temperatures. Shokr and Sinha [7] point out that for common metals, whose melting temperature is in the range of 3273K to 3683K, the density changes near the melting point may not have been examined precisely, suggesting that the same effect of increasing density may exist.

2.1.2 Types of ice

The shapes, sizes, and orientations of the individual crystals vary significantly, depending on the growth conditions. Shokr and Sinha [7] adapt the classification of Michel and Ramseier [8], which contains five main categories and up to five subcategories. The classification is summa-

rized in Table 2.1. In addition to the provided list, each subclass can have fine, medium, large and very large grain diameters, which are <1 mm, between 1 mm and 5 mm, between 5 mm and 20 mm and >20 mm, respectively.

Table 2.1: Categories and subcategories of freshwater and seawater ice types; reproduced from Shokr and Sinha [7].

Category	Subcategory	Description
Primary ice	P1	Calm ice surface with small temperature gradient: Grains are large to very large (can be more than 1 m) with irregular boundaries. The crystallographic orientation of c-axis is preferred vertical. Found only in calm water.
	P2	Calm ice surface with large temperature gradient: Grain size medium to very large. Crystallization progresses rapidly. Grain shape varies from tabular to needle-like and can be several centimeters in length. Crystallographic orientation is random or vertically preferred (freshwater ice only) superimposed on a random orientation. Dendrites are common.
	P3	Agitated surface nucleated from frazil: Grains are fine to medium with tabular shape. Crystallographic orientation is random.

	P4	Nucleation by snow: Grains are fine to medium and equiaxed, depending on snow type. Crystallographic orientation is random.
Secondary ice	S1	Columnar ice, preferred vertical c axis: Grains are perpendicular to the ice cover surface. Size increases with depth. Orientation of some crystals is preferred (natural selection growth). Grain size ranges from large to very large with irregular boundaries. Forms as a result of conditions given for P1 or P2 ice. Found in bodies of fresh water with low velocity. S1 type of ice is not sustainable in sea ice except for a thin layer, a few mm in thickness, at the top surface.

S2 Columnar ice, preferred horizontal c axis but random:

Conditions are the same as for S1 but primary ice has a random orientation. Cross-sectional (normal to long axis of the columns) grain size ranges from fine to large with irregular boundaries. Grain size increases more rapidly with depth than S1. Forms as a result of conditions given for P2, P3, and P4 types. This type is isotropic in horizontal plane, but it is columnar vertically. In material science this type of crystalline structure is known as traversal isotropic or orthotropic. S2 type columnar-grained material is also known as 'Directionally solidified' (DS) in the field of gas-turbine engine materials. This is a common ice type for level first year sea ice.

S3 Columnar ice, preferred aligned horizontal c axis: Forms at the bottom of thick freshwater lake ice sheet. In case of sea water, S3 type of ice is very common in areas with water current.

	S4	Congeaed frazil slush: Frazil is formed in turbulent water and is swept under the secondary ice cover. Grains are equiaxed and tabular with fine to medium size. Crystal boundaries are irregular and crystallographic orientation is random. Found in turbulent seawater.
	S5	Typically vertically oriented needle-shaped crystals: Created in turbulent water, frazil crystals are aligned and compacted by wave actions.
Superimposed ice	T1	Snow ice: Rounded to angular equiaxed grains with fine to medium size depending on the age of the snow. Formed when water saturates the snow deposited on ice surface, then freezes.
	T2	Drained snow ice: Randomly oriented but homogeneous rounded grains with fine to medium size rounded. Found where water level vary, thus draining a previously saturated snow that then freezes.
	T3	Surface ice: Layers of columnar ice formed on top of primary ice.

Random	R	Agglomeration of various ice types and forms. Ice growth process is disturbed by mechanical or other interference. Grain size varies from fine to very large. Crystal boundaries regular to angular and orientation can be random to preferred.
Glacier ice		Land-based ice masses: The origin of icebergs, bergybits, and the massive ice islands. Form as a result of snow deposition, compaction, and melting and freezing over hundreds of thousands of years.

Table 2.1 identifies many forms of ice according to the formation process, crystal type, and orientation. A simpler classification of freshwater ice would distinguish only two broad types – granular and columnar. (1) Granular ice consists of randomly oriented grains, whose sizes may vary, typically between 1 mm and 3 mm. Such type of structure usually grows from freshwater or develops from compressed snow in glaciers. This ice is easy to produce in the laboratory by flooding the crystals with water, so most laboratory studies are carried out on this type of ice. In some literature, it is called T-1 ice. Due to the random orientation of crystals, this material is isotropic. (2) Columnar ice has elongated grains with a preferred orientation, whose c-axes lie in the same plane. Individual crystals measure 3-100 mm. It is called S-2 ice. Such ice is found in the lower layers of a lake or river ice, and is orthotropic – its properties are the same only in the horizontal plane. In some cases, all crystals of columnar ice have their c-axes oriented in the same direction, making the material fully anisotropic.

2.1.3 Formation of sea ice

In freshwater basins, the freezing process does not involve convective flow. Once the temperature of the water goes below $+4^{\circ}\text{C}$, the density of the water will decrease as the temperature decreases. As a result, the cold layer of water at the surface of the basin is stable. Without the convective heat transfer, the freezing process is relatively fast.

On the other hand, the freezing of seawater is accompanied by the continuous convective flow. For the salinities higher than 24.7 parts per thousand (ppt) by weight, the density is monotonously increasing with decreasing temperature until the freezing point is reached [3]. As the surface water cools down, its density increases and the cold layer "sinks" to a certain depth, simultaneously being replaced by the warmer water from the lower layers. Only the upper 10-20 m of the basin is involved in this convective process, as the underlying water is stable due to higher salinity. The freezing point of water depends on the concentration and the types of solutes. For the standard seawater, with the salinity of 35 ppt, the freezing temperature is -1.91°C .

Once the water cools below the freezing point and the convective process stops, small ice crystals begin to form and float on the surface. The crystals, measuring up to several centimeters in diameter, are known as frazil ice. The c-axes are usually oriented vertically. As the freezing continues, the crystals become loosely linked together and form a stable layer of slush on the surface, known as grease ice due to its viscous appearance. The next stage involves the

formation of a solid cover, up to 5 cm thick. In the presence of waves, the cover breaks up into separate fragments, which collide and crush against each other. At certain conditions, the fragments become rounded disks, called pancake ice.

During the next period of development, the separate fragments solidify into a stable ice layer measuring 5 to 30 cm thick, known as young ice. It has a greenish-blue appearance [3] and is moist at the surface. The constituent pancakes or fragments can be discerned in the solid layer. The layer consists of granular ice, with individual grains measuring about 1mm in diameter and having random orientations due to crushing and mixing at earlier stages. The grains have high purity, as the solutes are rejected and expelled during freezing, although they become trapped between the grains in the form of fluid and gas pockets. The gross salinity of such ice is 5-10 ppt, the exact amount of brine and gas volume differs depending on the freezing conditions.

The thickness of the ice cover can be roughly estimated by an empirical formula, based on cumulative Freezing Degree Days (FDD), which is a measure of how cold it has been and how long it has been cold. The cumulative FDD is usually calculated as a sum of average daily degrees below freezing for a specified time period, e.g., since the onset of negative temperatures. The freezing temperature of ocean water is around -1.8°C . For example, if the outside temperature was -4°C for 20 days, the cumulative FDD is calculated as

$$FDD = (-1.8 - (-4)) \times 20 = 44$$

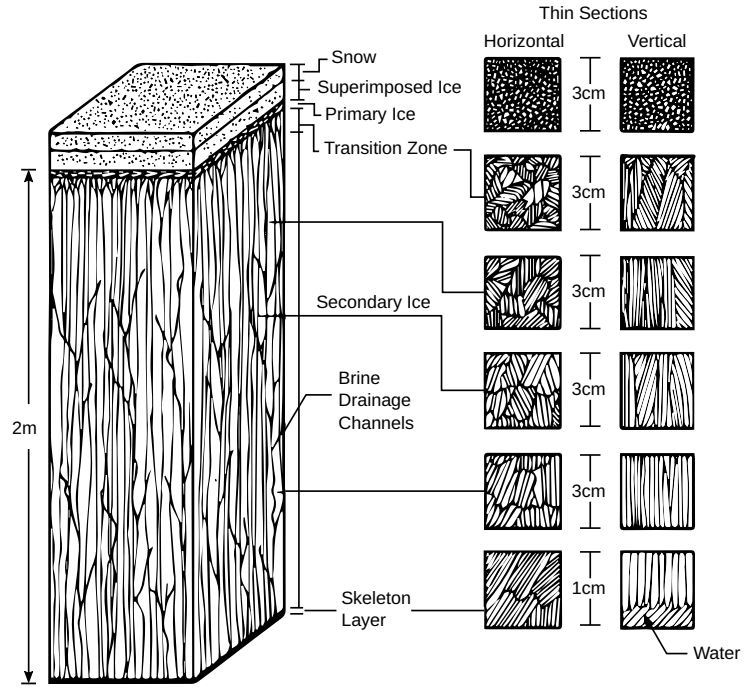


Figure 2.3: Structure of first year ice. From [10].

The equation to relate the temperature records to the growth of level ice h is

$$h = s(FDD)^n,$$

where the coefficients s and n depend on the units and geographical location. For the Beaufort Sea, for example, s is 0.025, with degrees measured on the Celsius scale, and the thickness measured in meters. The value of n is usually taken to be 0.5, as this fits the theoretical derivations [9]. Using these values, the thickness of ice, which corresponds to 44 FDD, would be

$$h = 0.025(44)^{0.5} \approx 0.17.$$

After a solid layer of young ice has formed on the sea surface, the growth continues at the

bottom of the ice sheet (Figure 2.3). Below the floating sheet, the water conditions are much calmer; therefore, the growth has a more ordered pattern. The growth speed is determined by local temperatures; typical speeds are on the order of 1 cm per day. Provided that the temperature of surrounding air remains constant, the growth rate decreases with time, as the formed layer of ice works as an insulator. The crystal structure is comprised of the larger crystals, measuring up to 10 cm and elongated vertically. The details of the formation of sea ice are discussed in Refs. [3], [11], [12].

2.2 Continuum behavior, elasticity, creep and constitutive models

Depending on the conditions of loading, polycrystalline ice Ih exhibits both brittle and ductile behavior. At common temperatures, the ice is brittle when uniaxial tension rate exceeds 10^{-7}s^{-1} , or when uniaxial compression rate exceeds 10^{-3}s^{-1} . At a lower rate, the behavior is ductile and exhibits inelastic strain greater than 1%. The presence of two slip systems explains some of the ductile behavior; however, that factor alone is insufficient to maintain the continuity of the grains [13]. Dislocations are a type of line defects in crystals. Dislocation glide causes one plane of molecules to slip over another, resulting in deformation (Figure 2.4). In the crystals of ice, the slip of the basal plane is responsible for the deformation, and slip on any other plane is difficult [14]. Such behavior is highly anisotropic, but in polycrystalline specimens, where grains are oriented randomly, deformation of multiple grains leads to isotropic macroscopic behavior.

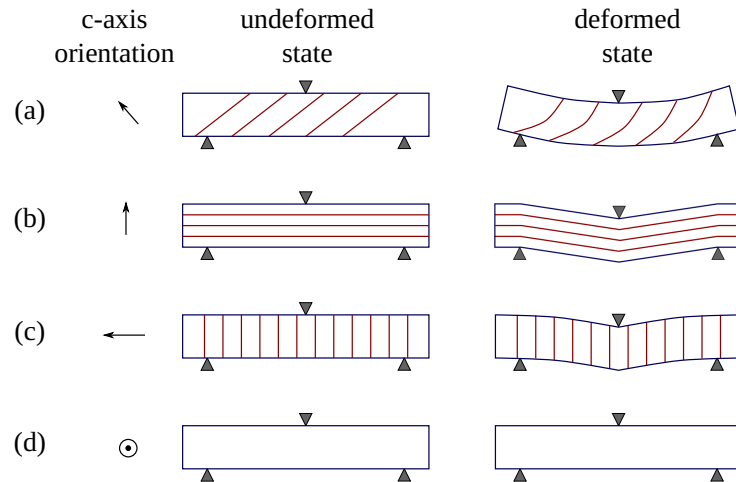


Figure 2.4: Plastic deformations in three-point bending of pure crystals with different orientations of c-axes. After [15].

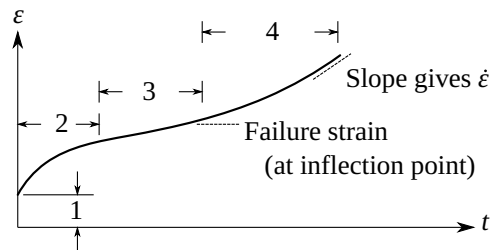


Figure 2.5: A typical strain-time curve for a constant strain stress test. Numbered stages are described in the text. After [16].

Upon application of constant stress to polycrystalline ice, the initial elastic deformation will take place, followed by the creep with decelerating rate, and ultimately the stage of accelerating creep rate. The causes of such behavior were discussed by Hooke et al. [16], Nadreau and Michel [17] and were subsequently summarized by Sanderson [3].

Figure 2.5 shows a schematic representation of the time-strain relationship for a sample under applied stress. Four stages correspond to different responses of the material, as indicated on the plot. The first stage is the immediate elastic response, sometimes denoted as ϵ_e . This response is usually almost linear (the strain is proportional to the applied stress), takes place instantaneously, and is attributed to the change of the distance between the individual molecules in the crystal lattice. The instantaneous strain can be described by Hooke's Law:

$$\epsilon_e = \sigma/E,$$

where E is Young's modulus, whose value for pure freshwater ice is approximately 9.5 GP. Young's modulus of individual crystals is between 8.6 GP and 12 GP, depending on the orientation of the force application [18]. Poisson's ratio ν for pure ice is approximately 0.33 ± 0.03 .

Whereas the instantaneous strain occurs almost immediately, subsequent stages indicate time-dependent behavior. At stage two, the delayed elastic strain ϵ_d and the viscous strain ϵ_v come into play. Stage three is characterized by the stabilization (reaching maximum) of the delayed elastic strain, whereas the viscous strain continues to increase. Stages two and three are known as secondary creep. The last phase of the continuous deformation is shown on Figure 2.5 as

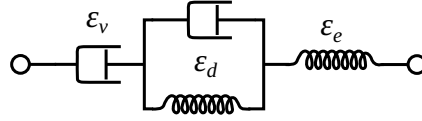


Figure 2.6: A spring-and-dashpot model of viscoelastic behavior of ice under uniaxial stress. Three sources of deformation are shown: instantaneous strain ϵ_e , delayed elastic strain ϵ_d and the viscous strain ϵ_v . After [19].

stage four and is known as tertiary creep.

This model of continuum behavior of ice can be represented schematically by springs and dashpots (Figure 2.6). The spring of the Maxwell unit represents the instantaneous strain ϵ_e , the dashpot of the Maxwell unit represents the viscous strain ϵ_v , and the whole Kelvin unit represents the delayed elastic strain ϵ_d . The total strain ϵ is the sum of the components:

$$\epsilon = \epsilon_e + \epsilon_d + \epsilon_v.$$

As soon as the load is applied, the delayed elastic strain ϵ_d starts to develop. This deformation is almost fully reversible – the sample will recover once the load is removed. Based on the experimental observations, Sinha [20], [21] suggested the following equation to quantify the amount of the delayed elastic strain as a function of time:

$$\epsilon_d = \frac{C\sigma}{Ed} \left[1 - e^{-(a_T t)^q} \right], \quad (2.1)$$

where d is grain diameter, E is Young's modulus, C is a constant [m], a_T is a function of temperature, and q is a dimensionless constant. Some experimentally determined values (for

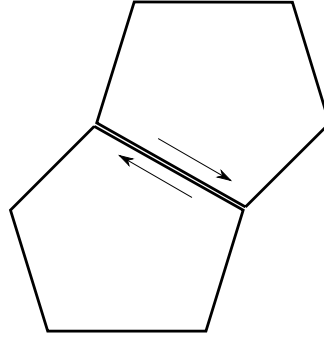


Figure 2.7: Schematic diagram of reversible grain boundary sliding. After [3].

columnar freshwater ice) are:

$$C = 9 \times 10^{-3} \text{ m},$$

$$E = 9.5 \text{ GPa},$$

$$a_T = 2.5 \times 10^{-4} \text{ s}^{-1},$$

$$q = 0.34.$$

The behavior of the delayed elastic strain is explained by its microscopic deformation mechanism, more precisely, by the sliding of boundaries between the grains. After the initial (immediate) elastic deformation, the grains become deformed, and stresses appear in the material. In the presence of stress, grain boundaries tend to slide via a mechanism of diffusional flow (Figure 2.7). The direction of sliding depends on the distribution of stresses and, ultimately, on the geometrical arrangement of the grains, which tends to remain constant (unless the sample fractures). Hence, grain boundary sliding is reversible, and no permanent deformation takes place. The reverse sliding also occurs by the diffusional flow.

At the end of stage two (Figure 2.5) ϵ_d reaches its asymptotic value $C\sigma/Ed$, and the defor-

mation process continues with the secondary (or steady-state) creep only. Secondary creep is an irreversible deformation within the grain, whose main mechanism is dislocation glide. It is present through stages two and three, with a relatively constant rate. An equation to estimate the rate was proposed by Glen [22]:

$$\dot{\epsilon}_v = B_i \sigma_{11}^n, \quad (2.2)$$

where n is a constant (approximately equal to 3), B_i is a function of temperature, and σ_{11} is the stress component along the loading axis. Models similar to equation (2.2) were described earlier, and are known as Norton-Hoff model for viscoplasticity. This model is not time-dependent, but in reality, secondary creep does not last indefinitely and eventually transitions to tertiary creep.

At stage four (Figure 2.5) the strain rate increases. Formation of microcracks is often observed, especially at grain boundaries, as dislocations tend to migrate towards the junctions of three grains. As the deformation continues, the cracks coalesce and lead to fracture. Tertiary creep is also possible without the formation of cracks if recrystallization occurs, i.e., the deformed grains being replaced by new undeformed ones, allowing the process to continue without fracture. This new structure may then deform again by delayed elastic strain. Dynamic recrystallization may occur repeatedly. A more detailed discussion of creep processes was done by Sanderson [3].

2.3 Fracture, strength, flaws, and ice-structure interaction

Fracture of ice occurs naturally in the environment or can be caused by human activity. At strain rates lower than 10^{-4} s^{-1} , the creep is prevalent, but at higher strain rates the fracture becomes the dominant process. Investigation of fracture properties of ice is essential for the engineering purposes, as the contact forces with offshore structures can exceed the levels at which the fracture begins. Structures, in this context, are bridge piers, lighthouses, channel markers, offshore platforms and other human-made objects that may have to withstand the forces of the oncoming ice. Sodhi and Haehnel [23] attempt to estimate the loading forces of the crushing ice sheet by analyzing the results of the small-scale and medium-scale indentation tests, as well as full-scale measurements. They note that the increase in the indentation rate leads to brittle failure mode, which, in turn, leads to a decrease in effective pressure. The forces depend on the temperature of the ice (strength increases when temperature decreases).

Environmental forces, which drive the motion of ice in the natural environment, can be quite large. When an ice sheet crushes against a structure, this failure process limits the forces that can be transmitted. The methodology of estimating the transmitted forces is usually based on analyzing the strength of ice and the way it interacts and fails. Palmer et al. [24] point out four fracture modes of an ice sheet interacting with an offshore structure or a natural obstacle. These modes are crushing, spalling, radial cracking, and circumferential cracking (Figure 2.8). (a) During crushing, the contact zone becomes highly fragmented and eventually turns into fine-grained ice particles. Cracks grow in all directions without preferred orientation. The

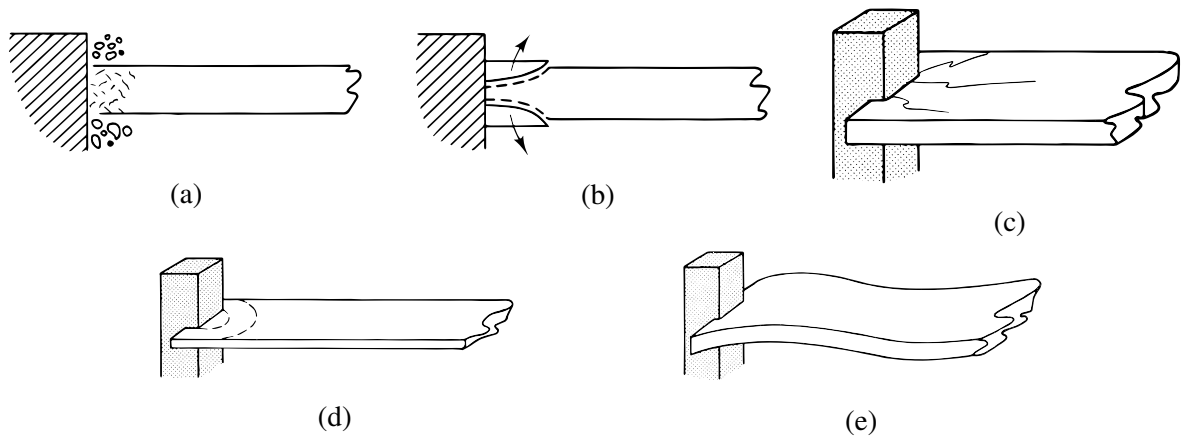


Figure 2.8: Fracture modes: (a) crushing; (b) spalling; (c) radial cracking; (d) circumferential cracking; (e) buckling. From [3].

dimensions of the process zone are of the same order of magnitude as the contact extent. Crushing is believed to cause the highest force on a structure [25]. (b) Spalling produces a smaller number of relatively large fragments that normally initiate with in-plane horizontal cracks that grow away from the zone of contact. The cracks reach either the upper or lower surface, and the fragments of ice break away either upwards or downwards. The typical size of the fragments is usually larger than the thickness of the ice. (c) Radial cracking occurs for high aspect ratios (indenter width D to ice thickness h), i.e., when the ice is relatively thin. Cracks propagate from the contact zone and cross the whole thickness of the sheet. The cracks have limited length and are initially stable. (d) Circumferential cracks occur as a result of buckling or out-of-plane bending, usually also at high aspect ratio D/h . The resulting fragments are often triangular or trapezoidal. The circumferential mode is often mixed with the radial model. (e) Sanderson [3] also points out a buckling mode, which is present in relatively thin ice. While buckling itself is not a type of fracture, it is a commonly observed unstable process that leads to the formation of circumferential cracks.

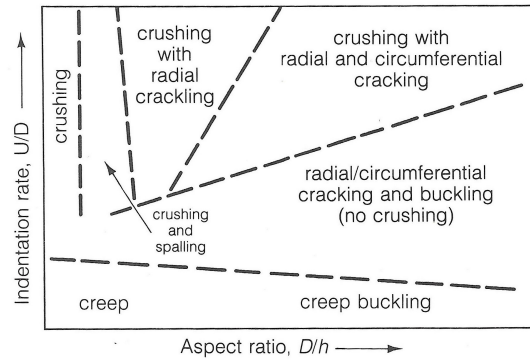


Figure 2.9: Deformation mode map of ice failure as a function of indentation rate and aspect ratio. The covered aspect range is approximately from 0 to 25. The indentation rate range is about 10^{-3} s^{-1} to 1 s^{-1} . From [3], based on [26] and [24].

According to Palmer [24], various modes may occur simultaneously, e.g., radial cracks developing away from the crushing zone. There is no definite rule that would predict the fracture mode based on indentation geometry. However, Sanderson [3] provides the following notes along with a schematic plot (Figure 2.9).

- for low indentation rates, creep is prevalent regardless of the geometry
- at low aspect ratios D/h , crushing mode is prevalent
- at higher aspect ratios, radial cracks form
- at low indentation rate and high aspect ratio, circumferential cracks may develop without crushing

It should be noted that Figure 2.9 does not have a scale on the axes, as the provided map is not intended to be precise. Other factors, such as temperature, salinity, and grain size, also play significant roles in determining the fracture mode and the overall strength of the material.

In the crushing failure mode, cyclic loading pattern is often present. This phenomenon is called ice induced vibrations and was first documented after the events at Cook Inlet drilling platforms in the 1960s. The accounts are presented by Peyton [27] and Blenkarn [28], who found that the ice failed by "ratcheting," now commonly known as "sawtooth" loading. Blenkarn noted that the sawtooth loading is self-excited and depends on the properties of both the structure and ice. Stiffness of the structure and its natural modes of vibration are the critical properties in the interaction mechanism. Different modes of loading exist, including a steady "lock-in" mode – a form of resonance.

Similar phenomena were registered at other flexible structures, such as lighthouses [29], bridges [30] and jacket oil platforms [31]. Ice-induced vibration is a hazard to offshore operations – it can lead to fatigue damage and reduce the work efficiency, as well as disturb the sleep of people working on the platform. There are documented cases when ice-induced vibrations lead to structural failure [32]. Since the early observations at Cook Inlet, there have been many experimental tests: some on sea ice, some on freshwater ice. According to Palmer [33], the mechanics of ice-induced vibration is not understood, and it is unclear how to transfer information between the model scale and the full scale. Questions exist about whether the small-scale models sufficiently reproduce the full-scale processes or if they are fundamentally different. Palmer [33] compares data from a variety of sources, both full-scale and laboratory-scale. The data are summarized in Table 2.2.

Table 2.2: Data sources. From [33].

First author	Full or model scale	Structure and loca- tion	Type of ice	Structure diameter (m)	Lowest frequency (Hz)
Jefferies	FS	Molikpaq, Beaufort	Sea	89	1.3
Yue [31]	FS	JZ9-3, Bohai	Sea	1.76	2.32
Yue [31]	FS	JZ20-2 MSW, Bohai	Sea	1.2	1.37
Bjerkas	FS	Nordstromsgrund, Baltic	Baltic sea (low- salinity)	7.5	2.34
Montgomery	FS	Athabaska river, AB	Fresh	2.32	8.9
Guo and Yue [34]	MS	(Laboratory)	Fresh	0.2	2.3
Guo and Yue [34]	MS	(Laboratory)	Fresh	0.12	2.3
Sodhi [35]	MS	(Laboratory)	Fresh	0.05	7.1 to 14.3
Maatanen	MS	(Laboratory)	Urea ice	0.1	7.5 to 24

Singh	MS	(Laboratory)	EG/AD/S ice	0.06	17.6
Sodhi [36]	MS	(Laboratory)	Fresh	0.1 and 0.146	8.3 to 15.1

2.4 Experimental techniques for measurement of ice properties

Certain types of tests of ice, especially the small-scale experiments, follow standard procedures, and are conducted regularly. For such tests, ample data are available. In turn, medium- and large-scale experimental data, such as the tactile sensor data collected by Japan Ocean Industries Association [37], [38], are scarce, and most of such experiments have unique geometries and other unique characteristics. Overall, a variety of experimental techniques is utilized to investigate the fracture behavior of ice. Many experiments are done in the labs, but in situ measurements are also performed on natural ice covers. This section does not aim to cover all experimental work on the mechanics of ice. Instead, several methods will be discussed to give the reader a general idea about the experimental techniques. Standardized testing methods can be found in [39], and were also described in International Towing Tank Conference (ITTC) Recommended Procedures for testing ice properties.

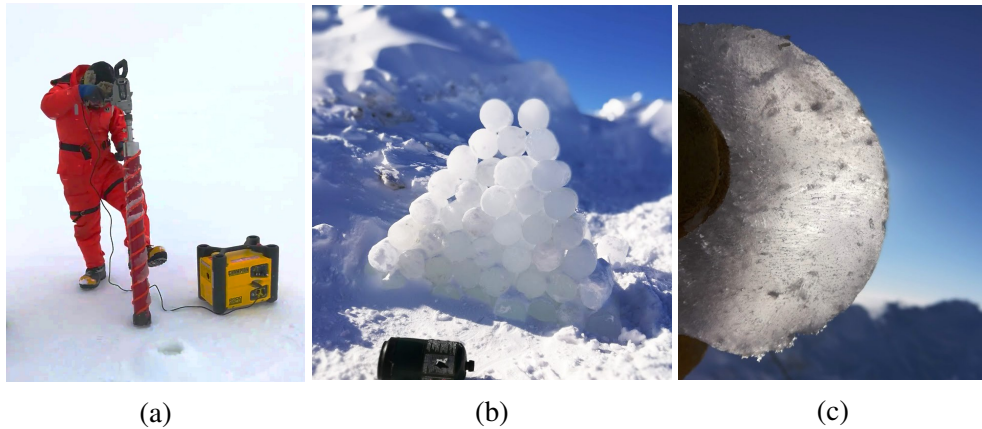


Figure 2.10: (a) Extraction of a core sample from the frozen cover; (b) Fractured samples after testing; (c) Bubbles and other inclusions in a sample.

2.4.1 Small-scale tests

The use of cylindrical samples for small-scale tests is widespread because of the simplicity of their preparation. In the field, such samples are extracted using a drill (Figure 2.10a), whereas, in the lab, such samples are prepared by turning rectangular blocks on a lathe. Common sizes are 8-10 cm in diameter and 15-30 cm in length (Figure 2.10b). To investigate the mechanical properties of ice, cylinders can be subjected to uniaxial compressive tests, as described in Refs. [40]–[42] or uniaxial tensile tests [43]. In a more recent approach, the digital speckle correlation method is used to measure deformation on the surface of the sample during the loading process [44]. Asymmetric Four-Point Bending test (AFPB) [45] measures the shear strength of the sample. Other types of tests include high-velocity impact fracture [46]–[49], Brazilian tests and experiments with various degrees of confinement [50]. Samples of natural ice, even from the same location, may vary significantly in the content of inclusions (Figure 2.10c) and, consequently, in their mechanical properties.

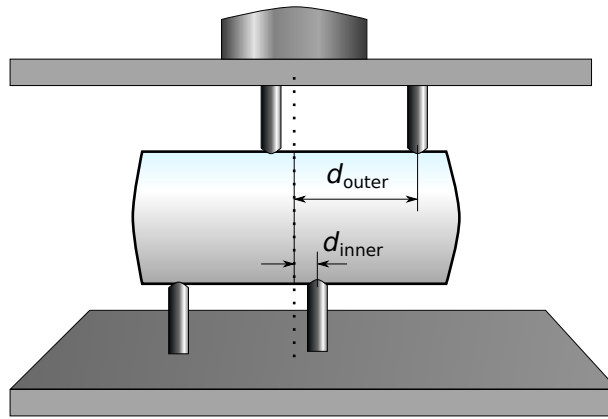


Figure 2.11: Schematic representation of the AFPB machine. Ice sample (in the center) is fixed between four loading pins that move vertically. Torque is applied through the pins creating a zone of high shear stress in the middle. After [45].

The uniaxial compressive strength of ice is measured by loading a cylindrical sample to the point of fracture. A complete force-time (or force-strain) relationship is usually recorded. The measured force is converted to pressure by dividing it over the area of the cross-section of the sample. Typical strengths for freshwater samples vary between 7 MPa and 10 MPa [41]. Young's modulus can be evaluated from the obtained data by inspecting the slope of the initial linear portion of the loading curve. Typical recorded values for freshwater laboratory ice are on the order of 10 GPa at -10°C . Finally, shapes of the fractures can be inspected in the cases when distinct fracture paths are present.

AFPB test measures the shear strength of the sample by applying the load through four asymmetric pins. The inner pins create a static torque in one direction, whereas the outer pins apply the same torque in the opposite direction. As a result, the sample remains immobile, and an area of high shear stress is created between the inner pins. Once the fracture occurs, peak force

P is recorded and converted to peak shear stress using the formula:

$$\tau_{max} = \frac{4}{3} \frac{(1 - \alpha)P}{(1 + \alpha)\pi r^2},$$

where r is the radius of the cylinder. Parameter α is defined as:

$$\alpha = \frac{d_{inner}}{d_{outer}},$$

where d_{inner} is the distance between the center of the inner pin and the center of the cylinder (loading axis); d_{outer} is the distance between the center of the outer pin and the center of the cylinder (Figure 2.11). Typically measured shear strengths for solid seawater ice are between 400 kPa and 750 kPa [45].

Schulson [41] investigated the relationship between the strength of a sample under uniaxial compression and its grain sizes. It was experimentally observed that larger grains correspond to weaker ice following the Hall-Petch equation [51]:

$$\sigma_y = \sigma_0 + \frac{k_y}{\sqrt{d}}, \quad (2.3)$$

where σ_y is the yield stress, σ_0 is a constant for the starting stress for dislocation movement, k_y is the strengthening coefficient (a constant specific to each material), and d is the average grain diameter.

Small-scale laboratory tests also include various indentation setups that differ by geometries

of the sample and the indenter, degrees of confinement, indentation rates and types of ice being tested. Aside from determining the strength of the material, indentation tests evaluate the formation of a layer of microstructurally modified material under the indenter. These tests help to understand the mechanics of ice failure and the formation of high-pressure zones during ice-structure interaction. Indentation processes are usually accompanied by extrusion of the crushed ice from beneath the indenter and cyclic loading patterns, whose frequency and magnitude depend on the indentation rate. Descriptions of unconfined tests are presented, for example, in Refs. [26], [52], [53].

In the natural environment, ice-structure interaction is often accompanied by high degrees of confinement near high-pressure zones. Additionally, structural compliance has a significant effect on the interaction processes and associated loads. Experiments attempt to reproduce these situations and investigate the roles of scale, temperature and structural compliance on dynamic ice-structure interactions [54]–[58].

2.4.2 Full-scale observations

The process of crushing of ice against a vertical structure is very different from the controlled failure that occurs in laboratory testing on uniaxially loaded cylinders. Alongside with crushing, ice-induced vibration was recorded in many field observations, especially for fixed structures. It is impossible to estimate the pressures on offshore structures by evaluating the laboratory data alone because there are differences in the scales of the processes. For this reason,

full-scale data is valuable to researchers and engineers. A summary, in the context of ice-induced vibrations, is presented in Table 2.2. Palmer and Croasdale give another good summary in their book [9], where 30 sources of data are listed. These data were obtained between the years 1963 and 2003. The listed sources are well-known to the ice research community, and include Confederation Bridge in Canada [59], [60], Nordstromsgrund lighthouse in Sweden [61] (LOLEIF- and STRICE-Projects), Molikpaq measurements in Russia [62], [63] and in the Gulf of Canada [64], [65], Bohai Bay measurements in China [31] and iceberg impacts at Grappling Island in Labrador [66], [67]. Another set of observations comes from the drill barge Kulluk that was operational in Canadian Arctic [68].

According to Palmer and Croasdale, the following measurement methods are applied for ice loads on full-scale structures:

- Structure response using strain gauges
- Foundation response using soils instruments
- Structure response using extensometers
- Structure response using accelerometers
- Structure/foundation response using tiltmeters
- Ice load cells and panels at the interface between ice and the structure
- Instrumenting the surrounding ice for strains or pressures

- Estimating ice loads from the deceleration of free floating floes impacting the structure.

The latter method is quite elegant, as it allows to deduce ice forces from an isolated floe impact by merely observing the motion of the floe from a distance. Such an approach was performed at Hans Island in 1980, 1981, and 1983. The impact forces are calculated based on the measured decelerations of the floes and their estimated mass [69].

Extensometers and strain gauges measure strain on the object. The technology appeared as early as 1938, and with contemporary computers, multiple channels of data can be recorded. When using extensometers, one has to keep in mind that complex structures have multiple load paths, i.e., nearby measurements can differ significantly. Finite element analyses sometimes help to evaluate the load distribution on the whole structure or to calibrate the gauges; however, such FEM calculations usually work with idealized representations that consider only structural elements and may give inaccurate results. Despite various impediments, strain gauges and extensometers are useful tools for recording full-scale data.

When *tiltmeters* are placed at several different spots of the structure, they can give angular deflection of one point relative to another, as well as the total deflection of the structure. They work best on slender structures, such as bridge piers, and were installed, for example, on the Confederation Bridge.

Accelerometers are considered unreliable because structures do not behave as simple mass-spring systems. They may be used as an additional tool for recording vibrations.

On the other side of the spectrum of the equipment are *load cells* and *panels*, which measure local pressures at the points where they are installed. The data obtained from such panels influenced current ice design criteria [9]. Combining the data from several adjacent panels gives full loads or, at least, forces over the area where they are deployed.

Finally, monitoring the response of the foundation can give an insight into high ice loads. Properties of soil and the mechanical properties of the structure must be considered when evaluating the results. Short peak forces may not register with this method.

2.4.3 Medium-scale tests

Medium-scale experiments attempt to overcome the scaling issues of the laboratory measurements and get as close as possible to reproducing the studied processes in a controlled setting. They are usually conducted in the field with the samples tested in situ. Examples of such tests include measurements of in situ beam bending tests (to measure flexural strength) [45], [70], with the dimensions of the samples being up to several meters long and over one meter wide. Other tests include indentation and impact experiments with the forces applied vertically [71] or horizontally [72].

Valuable results were obtained during the Medium Scale Indenter (MSI) Tests – a series of ice load experiments conducted in the Canadian Arctic [73]. The first of these experiments were done in 1984 at Pond Inlet, Baffin Island [74]. Iceberg indentation tests were performed with a large spherical indenter driven by four hydraulic actuators, each with a 4 MN capacity. The

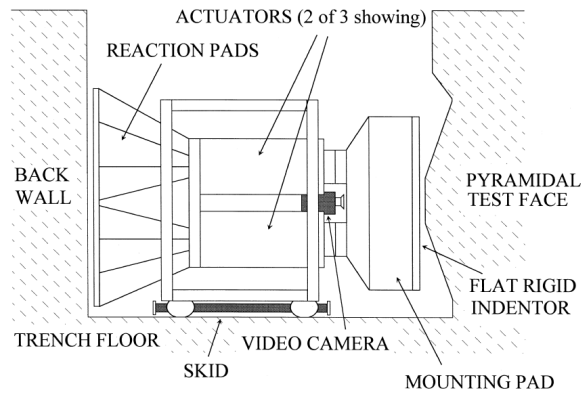


Figure 2.12: Schematic of the large indentation apparatus in the trench. From [72].

equipment was later re-purposed for another series of tests that took place in May of 1990. Medium-scale indentation experiments were performed on the multiyear sea ice at Hobson's Choice Ice Island [72], [75] using an indenter outfitted with a window and video camera so that the ice behavior could be observed (Figure 2.12). The hydraulic actuators were equipped with load cells allowing to record the loading curve. Additionally, pressure and temperature were monitored at the face of the indenter. From the obtained data, researchers estimated the following related quantities: the peak stress, elastic strain, delayed elastic strain, and permanent viscous creep. These values were subsequently compared with other types of ice and with laboratory measurements. In the laboratory, small-scale uniaxial compression tests were performed on the samples from the main test site using both undamaged and damaged ice.

Another set data comes from Japan Ocean Industries Association (JOIA), whose medium-scale field indentation tests (MS-FIT) provided high-resolution data from tactile pressure sensors [76], [77].

Between 1993 and 1995, the following tests were performed [76]:

1. Mechanical properties confirmation tests of model ice (1993). The compressive strength, bending strength, and other mechanical properties of granular and columnar model ice used for the following tests were confirmed.
2. Small indenter test (1994). To clarify the ice failure mechanism and the relation between indentation velocity and crushing depth, small indenter tests were conducted. In the test, the indenter was pushed against the vertical ice sheet in the air to facilitate observation.
3. Indentation tests by stiff rectangular and circular models (1993). Indentation tests were performed using four different size rectangular models of 100 mm, 400 mm, 800 mm and 1,500 mm in width and one circular model with a diameter of 800 mm. Segmented local ice pressure panels were installed to measure local ice pressure characteristics.
4. Compliant rectangular model indentation tests (1994). To investigate the influence of structure stiffness, indentation tests were performed using a compliant 800 mm wide rectangular model. The model has segmented local ice panels in the water line and disk springs at the bottom to vary stiffness in three levels. Ice thickness ranged from 60 mm to 94 mm, and the compressive strength varied from 115 KPa to 300 KPa. The tests were performed with five different speeds ranging from 6 mm/s to 105 mm/s.
5. Indentation tests in ridge ice (1995). To investigate the characteristics of ice load in ridge ice, indentation tests were conducted with three different stiffness values.

Another set of data was recorded between 1996 and 2000 in sea ice at the harbor of Lake Notoro, Hokkaido. A segmented indenter was pushed against edges of floating ice sheets.

Each segment of the indenter contained a load cell to measure forces in three directions and a moment about horizontal axis [77].

The data recorded by JOIA became particularly valuable because it allowed to relate local pressures and global pressures and to investigate the role of high-pressure zones in ice-structure interaction [78]. Observations from the tactile pressure sensor suggest that the combined area of all hpzs during crushing events comprises about 10% of the nominal interaction area [79]. A more detailed discussion of the medium-scale tests is presented in [73], [75], [80], [81].

2.5 High-pressure zones, the mechanics of ice-structure interaction

During ice-structure interaction, compressive failure manifests in spalls, splits, and crushing, with the forces being transmitted in localized areas (Figure 2.13). Crushed ice, as well as larger spall fragments, may accumulate in large quantities. For example, piles of crushed ice up to 8 m high were observed around Molikpaq [64]. The crushing process is usually cyclic, with new spalls occurring periodically and leading to fluctuations of pressure.

According to Jordaan [82], there are two main situations that describe ice fracture during ice-structure interaction: (1) when the fractures are not specially induced, and the cracks run to closest free surfaces; (2) where fractures are induced, such as in the case with icebreaking ships and conical structures that cause flexural failure of oncoming ice sheets. Brittle failure occurs at speeds of a few millimeters per second and higher; lower interaction rates usually

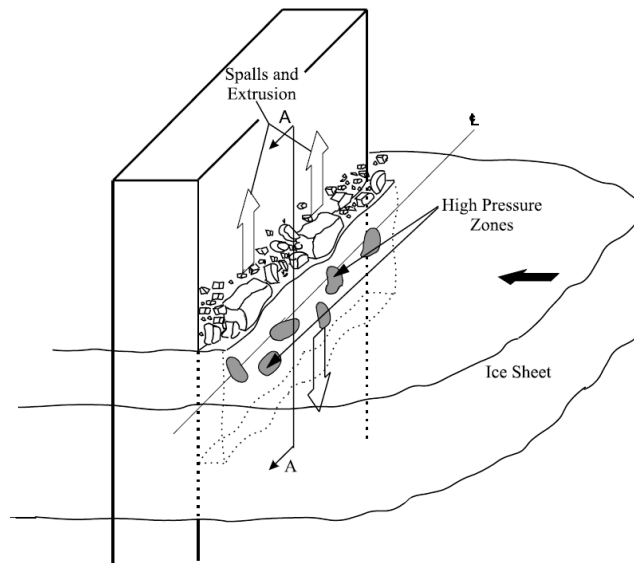


Figure 2.13: Illustration of spalling, crushing, extrusion and formation of hpzs. From [82].

result in ductile/viscous behavior (possibly with microstructural change). In the areas where compressive forces are transmitted from the structure to the ice, the pressures fluctuate. The locations of such areas fluctuate as well, depending on the path of crack propagation, which, in turn, follows the pre-existing flaws that are distributed randomly. The regions of force concentrations are called high-pressure zones or *hpzs*. The state of stress within *hpzs* is triaxial, with the highest pressures up to 70 MPa existing at the center of the contact. *Hpzs* tend to be located in the most confined regions, away from free surfaces.

The presence of high-intensity shear stresses in *hpzs* results in a microstructural change of ice in these areas, which, in turn, makes it more compliant in comparison with undamaged ice. *Hpzs* contain "highly recrystallized material," whereas near the edges of *hpzs* there is microfractured material where recrystallization also occurs.

Given the random nature of spalling events, as well as the natural variability of size and fre-

quency of ice features in the ocean, the approach to the design of offshore structures must include probabilistic analysis. For example, Fuglem et al. [83] developed a probabilistic framework for determining possible impact loads from icebergs for design purposes of offshore systems. An essential consideration in the design is local pressures. For small areas, these pressures can be quite significant, up to 70 MPa, but for larger areas, they are smaller. Average pressures are also low, on the order of 1 MPa. Jordaan [82] relies on the experimental data to show that this pressure-area relationship is determined by the fracture processes and the random distributions of the size and spatial density of flaws.

The failure stresses of the individual high-pressure zones are determined by the strength of its surrounding material, i.e., the layer at the boundary of the hpz. This quantity is also random, and the failure of individual hpzs can be interrupted by spalling events. Faster interaction rates lead to greater stress concentration and consequently, more spalls. At high rates, "continuous crushing" was observed, rather than load oscillations. The reader is referred to the work by Singh et al. [84] for a discussion about the flow properties of crushed ice. Additional investigations of the role of high-pressure zones in the fracture processes can be found in Refs. [56], [85]–[88].

2.6 Probabilistic analysis

Random events are present in many aspects of ice-structure interaction. For example, a certain region may be prone to the presence of ice, but the exact amount and the type of ice depend on

the long history of its formation and travel. Icebergs, for example, travel with wind and water currents, which vary in their direction and speed. Icebergs have irregular shapes and sizes, ranging from bergy bits and growlers of 1 m in size to the largest recorded iceberg B-15 that measured 295 km long and 37 km wide. B-15 calved in Antarctica in March 2000, and as of 2018, its fragments were still being tracked [89].

While some areas are more prone to ice conditions than the others, the concentration of ice cover can be quite unpredictable. In Canada, latest ice conditions are being tracked by Canadian Ice Service, whose mission is to provide the most timely and accurate information about ice in Canada's navigable waters. Real-time tracking is needed for safe and efficient maritime operations and to help protect the environment. The structure of ice sheets is highly irregular as well. The field measurements discussed in Chapter 7 show that the strength of the ice, for example, varies substantially even at the same testing site. This variability is due to the haphazard growth process in nature, the presence of brine, air pockets, and other inclusions.

On the other side, human-made structures that are built to sustain ice forces may have certain random aspects. For example, the materials from which these structures are built vary in their quality and strength. Depending on the type of interaction and the point of load application, structures may behave differently.

2.6.1 Pressure-area relationship

While there is much randomness in ice-structure interaction, it does not mean that the situations are entirely unpredictable. An early work that applies probabilistic analysis was done by Kry [90] and concerned the prediction of ice forces acting on a stationary structure, such as a bridge pier. Kry assumed that the peaks of pressure occur randomly in time at different zones of the contact area, so he divided the contact area into different zones. He used data from indentation tests to simulate a pressure history on one zone with an area equal to that in experimental data. The simulation was done for each zone, and the average pressure over the total area was calculated. Assuming that pressures on individual zones are independent, a relationship between the number of zones and the peak pressure is inferred: the larger the number of zones, the lower the probability that the pressure will simultaneously peak on all of them. The average pressure will decrease as the number of zones increases.

The pressure-area relationship was confirmed by observations (Figure 2.14), and a theoretical curve was proposed by Ashby et al. [91]. The data come from a variety of sources, including indentation experiments in the lab (small contact areas), medium-scale field tests that used ice blocks ranging between 1 m and 5 m in size. Finally, full-scale observations were made on Hans Island, which is approximately 1.7 km long and 1.3 km across. Palmer and Croasdale [9] give details about the data sources and the relevant discussion. The last group of data with the largest contact area comes from oceanographic numerical models (and is not very reliable).

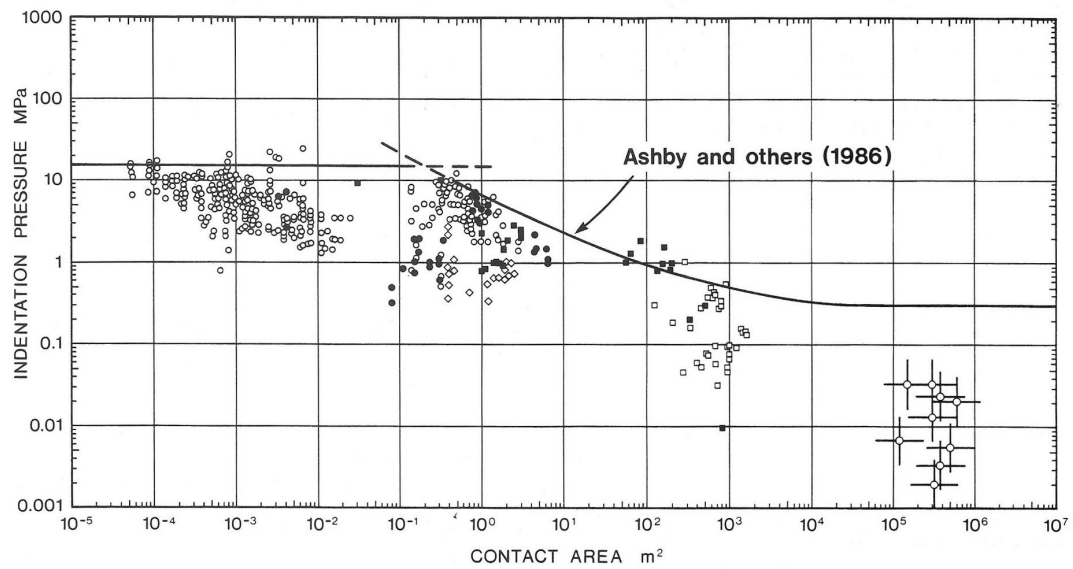


Figure 2.14: Theoretical pressure-area curve of Ashby et al. [91] constrained to the observed data. From [3].

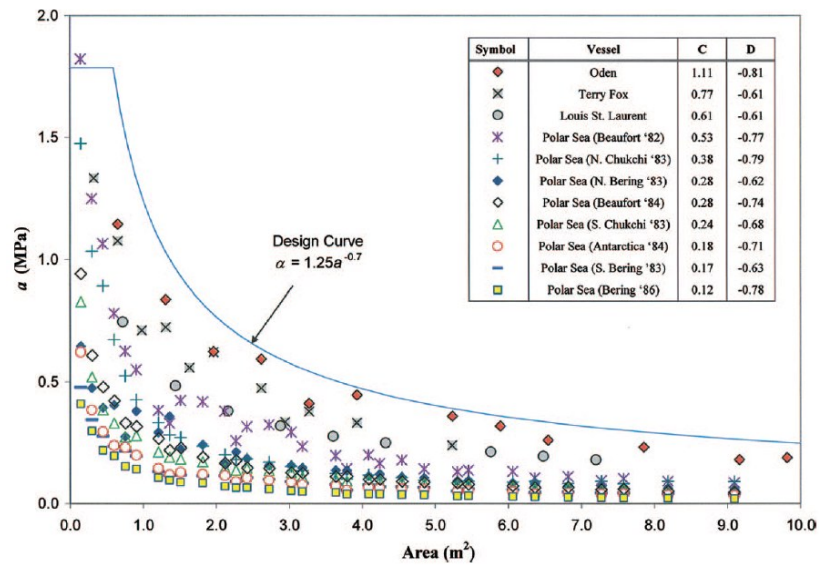


Figure 2.15: Pressure vs. area for ship-ice interaction. From [92].

For a smaller range of values, from zero to 10 meters, pressure-area data is less scattered (Figure 2.15). The shown data was obtained from the CCGS Louis S. St. Laurent, USCGS Polar Sea, Swedish Icebreaker Oden, and the CCGS Terry Fox. Local ice pressures were obtained mainly from strain gauge data. Figure 2.15 shows that the pressures follow a decreasing trend with the increasing area, and each data set has a distinct, well-defined curve. This relationship suggests that the pressure is dependent on some physical characteristic of the interaction, such as the ice type, thickness, or temperature [92]. The following equation is suggested to model the pressure-area relationship:

$$\alpha = Ca^D,$$

where α denotes pressure (MPa), a is area (m^2), and C and D are fitted parameters. Best fits of parameters for each data set are shown on Figure 2.15. The design curve with parameters $C = 1.25$ and $D = -0.7$ predicts the values of pressure that are not exceeded in the experiment.

2.6.2 Distributions for hull pressures

Approaches that involve probabilistic analysis are not limited to the pressure-area relationship. Parameters with uncertainty are: ice concentration and thickness, seasonal variability, speed of the ice, mass and added mass of the ice feature and vessel, speed of the vessel, interaction geometry, material properties including the density of random flaws [93]. An essential consideration in ship design is the distribution of ice loads experienced by a ship hull when traveling through a field of broken ice. The recorded experimental data typically follows exponential or

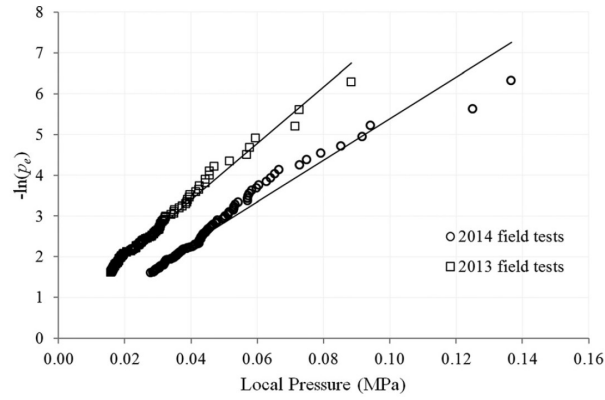


Figure 2.16: Comparison of local pressure for impact events on the bow shoulder measured in 2013 and 2014 field tests. From [94].

Weibull's distributions. The work by Jordaan et al. [85] shows a statistical analysis of local ice pressures. The experimental data were obtained from MV CANMAR Kigoriak and USCGC Polar Sea.

The pressure data used in the Polar Sea analysis consists of 513 impacts with multiyear and heavy first-year ice features (recorded by a panel in the bow of the ship). The data was presented in the form of a quantile-quantile (Q-Q) plot and compared against Weibull distribution. The pressure data from Kigoriak was obtained primarily in thick first-year and second-year ice. The recorded data consisted of 397 interaction events. The data was also presented as a Q-Q plot and compared against exponential distribution. The analysis allowed determination of exceedance probabilities of the design coefficients for pressure as a function of the area.

The work by Rahman et al. [94] follows along the same lines, analyzing the data from a totally enclosed motor propelled survival craft (TEMPSC) in controlled pack ice conditions. The trials were performed in a freshwater lake in a pool cut in the surrounding level ice.

The dimensions of the pool were approximately 80×35 m. Several data sets were recorded, including load distributions for stem and bow shoulders during the tests in 2013 and 2014. The Q-Q plot on Figure 2.16 shows the ranked pressure readings for two trials. The recorded data follow exponential distributions well. The work also analyzed the factors that influence ice loads, i.e., the average ice concentration does not have a significant effect on the magnitude, but the local variation of ice concentration does.

Suyuthi [95] states that on some occasions, the process of ice induced loads on ship hulls cannot be modeled by traditional statistical models. The exponential distribution is a single-parameter model, whose cumulative distribution function is given by the following equation:

$$F(x; \lambda) = 1 - \exp(-\lambda x),$$

where λ is a parameter. In turn, Weibull distribution has two parameters and is described by the following equation:

$$F(x; \theta, k) = 1 - \exp \left[- \left(\frac{x}{\theta} \right)^k \right],$$

where θ and k are parameters. Instead of using these models, Suyuthi suggests a three-parameter exponential model that improves the quality of fitting. The proposed model is a linear combination of two one-parameter exponential models, and is formulated as follows:

$$F(x; \lambda_1, \lambda_2, a) = a(1 - \exp(-\lambda_1 x)) + (1 - a)(1 - \exp(-\lambda_2 x)),$$

where a , λ_1 and λ_2 are parameters. Suyuthi includes a discussion of the applicability of this model based on measurements of ice-induced loads obtained from KV Svalbard during the winter of 2007.

2.6.3 Other applications of probabilistic analysis

Probabilistic analysis can be applied to other aspects of ice mechanics. For example, analysis of encounter probability with ice floes is presented by Sanderson [3] and Ralph [93]. A probabilistic approach to analyzing the high-pressure zones is presented in Refs. [96], [97]. By re-evaluating the data from JOIA, individual high-pressure zones are extracted and tracked. Statistical analysis of the extracted data allows, among other things, to create distributions of the area occupied by hpzs, analyze the depths of hpzs, their total volume, and perform the analysis of peak forces.

2.7 Other approaches to describing fracture

The previous sections considered such processes as the development of high-pressure zones, pressure softening, and strength measures of ice. While probabilistic methods take into account these aspects of ice behavior, they are built upon the statistical data collected in the field, or the empirically-established relationships between different parameters, e.g., interaction area and pressure. In a way, the physical details of fracture mechanics are not important

for the probability-based analyses. A different approach to investigating the fracture behavior is by constructing models that describe the details of the fracture process based on fundamental physics. Some of these models are discussed in this section. For a comprehensive theoretical analysis of ice failure, the reader is referred to Sanderson [3].

2.7.1 Wing crack models

Ice failure in compression is remarkably different from that in tension. Under tensile load, once a crack is nucleated, it usually propagates instantly to the free surface of the material, e.g., during spalls and circumferential cracks. In other cases, such cracks propagate to the areas where tensile stresses are no longer present, as seen in radial cracks. In compression, however, nucleated cracks reduce stress concentrations, and the object can sustain more deformation without fracturing. While propagation of cracks in tension is unstable, propagation in compression is stable (the process stops once the strains redistribute). The failure of the sample occurs by the linkage of a large number of cracks, not via propagation of a single crack. Failure in compression is a more complex process. It was experimentally observed that compressive strength is higher than tensile strength.

Cole [40] determined that the typical length of nucleated cracks $2a$ is proportional to the grain size d and obeys the relationship $2a = 0.65d$. The concentration of cracks also depends on the grain size; for larger grains, the concentration is higher; for grains exceeding 5 mm, there is about one crack per grain. About 90% of the cracks are oriented within 45° of the principal

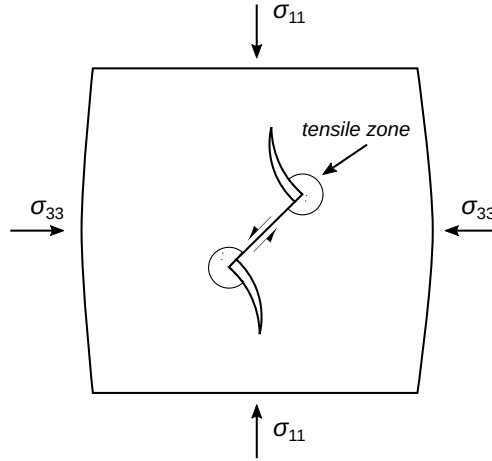


Figure 2.17: Formation of wing crack in a brittle solid under compression.

axis of compressive strength. The average angle is 23° with a standard deviation of 17° .

The two-dimensional model (Figure 2.17) shows how a single crack develops and propagates. Under the application of the principal stress σ_{11} and confining stress σ_{33} , two grains slide relative to one another, creating two tensile zones at the boundaries. In these tensile zones, the cracks form perpendicularly to the sliding direction, and as they grow, they align with the direction of the principal stress. A detailed analysis of the growth process of wing cracks and the fracture toughness of such material was performed by Ashby and Hallam [98]. Their model gives a reasonably accurate prediction of the ice compressive strength. The model [98] was also implemented in probabilistic framework for analysis of compressive failure during edge indentation by Taylor [78], who linked these processes with pressure-area scale effects.

Kolari [99] extends the work of Ashby and Sammis [100] by developing a 3D anisotropic continuum damage model for granular brittle solids. The model is implemented in finite element software ABAQUS, and numerical simulations are performed in the context of ice mechanics.

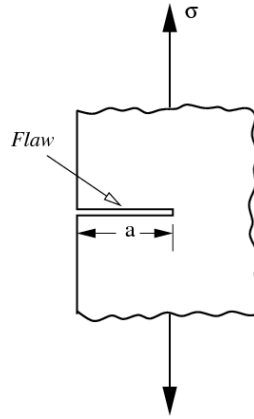


Figure 2.18: A cracked structure with initial notch of length a under stress σ .

Simulation results agree with the experimental data both under uniaxial compression and tension, and the effects of grain size and temperature are captured. The model was able to predict qualitatively and quantitatively the brittle failure modes and strength.

2.7.2 Linear elastic fracture mechanics

A simple theory that describes the propagation of cracks in the materials is Linear Elastic Fracture Mechanics (LEFM). The underlying assumption is that the material is isotropic and linear elastic. Based on the assumption, the stress field near the crack tip is calculated using the theory of elasticity. When the stresses near the crack tip exceed the material fracture toughness, the crack will grow. The success of this theory explained by its simplicity for theoretical, numerical, and experimental analyses of fracture.

Suppose that a plate of uniform thickness t is loaded in tension with stress σ . The plate contains a pre-existing flaw of length a (Figure 2.18). The strain energy density is $\sigma^2/2E$,

where E is Young's Modulus. The formation and propagation of a crack consume a certain amount of energy. As a result of propagation, some stress is released. Let us assume that during its formation, the crack releases all energy in a semi-circle centered at the end of the cracks (of radius a), and no other energy gets released. The energy release rate (per propagation distance) is then

$$\frac{d\psi}{da} = \frac{\pi\sigma^2 at}{2E}.$$

The growth of crack by an infinitesimal distance da creates two new surfaces with areas $t da$. If the energy required to create the surface is G (per unit area), then the energy consumption rate is

$$\frac{d\tilde{\psi}}{da} = Gt.$$

The crack will grow if the released energy is greater than the energy required for surface formation:

$$\frac{\pi\sigma^2 at}{2E} > Gt,$$

or

$$\sigma > \sqrt{\frac{2EG}{\pi a}}. \quad (2.4)$$

Taking into account that some energy is released outside of the semi-circle of radius a , equation (2.4) becomes:

$$\sigma\sqrt{\pi a} > \sqrt{EG}. \quad (2.5)$$

The left-hand side of equation (2.5) is called *stress intensity* and the right-hand side is called

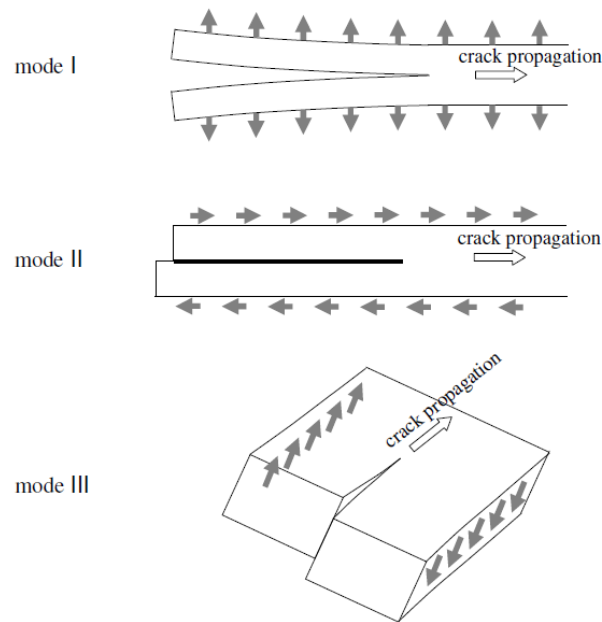


Figure 2.19: Fracture modes. From [9].

fracture toughness and denoted as K_{Ic} . The subscript c stands for critical, and I stands for the first mode of fracture (Figure 2.19).

The linear-elastic fracture toughness of a material is determined from the stress intensity factor at which a thin crack in the material begins to grow. The units are $\text{Pa}\sqrt{\text{m}}$. Fracture toughness is a quantitative way of expressing a material's resistance to brittle fracture when a crack is present. A material with high fracture toughness may undergo ductile fracture as opposed to brittle fracture.

2.7.3 Energy-based approach to understanding the fracture process of ice

One of the widely accepted measures of ice strength in engineering is the uniaxial compressive strength. This measure finds its way into the design of structures and ships that need to withstand a potential interaction with ice features. The compressive strength is determined experimentally by applying an increasing force to a small sample. The samples usually have a cubical or cylindrical form. Another related measure of resistance to fracture is fracture toughness – one of the most important properties of the material for design applications. The linear-elastic fracture toughness K_{Ic} is determined from the stress intensity factor. This value characterizes the consumption of energy needed for crack propagation, but this theory is mostly developed for pure homogeneous materials, whereas natural ice comes with inclusions, pores, and irregular defects.

Tsuprik [101] suggests that the reduction of the kinetic energy of a moving ice floe (or other feature) during ice-structure interaction can be approximated as the energy used for the fragmentation process during failure. He denotes the minimum amount of energy used per volume of fractured material as ε_{cr} – the specific failure energy. Based on theoretical and experimental studies, the author concludes that the specific failure energy is independent of test conditions, although its value depends on temperature.

In the subsequent related study, Bekker et al. [102] suggest experimental methods of measur-

ing the specific fracture energy of samples, whose form is trapezoidal beams and wedges. The mechanism of energy transfer from the ice field to a structure is discussed by Tsuprik et al. in [103]. Sometimes the process of ice destruction due to the contact with structure has cyclic character. During each cycle, the load variation acting upon the structure consists of repeated stages. Tsuprik et al. break apart one such cycle into three basic stages and different independent continuous processes. For more details about the energy-based approach, the reader is referred to Refs. [101]–[103].

2.7.4 Floe impact on a structure

Sea ice undergoes a transition from ductile to brittle behavior when (1) stresses exceed the fracture strength of the material, (2) strains exceed the value of about 0.01, or (3) strain-rates reach the level of approximately 10^{-3} s^{-1} [1], [3], [104]. Staroszczyk [105] points out that during this transition, the loads exerted by floating ice on a structure are at near-maximum values. Further increase in the strain rates, associated with the process of crack formation, does not increase the forces. The brittle failure takes place only at several relatively small regions of the ice–structure interface. The total exerted loads are highly irregular, both in time and location. The fracture process appears as characteristic sharp spikes dispersed at irregular time intervals during measurements.

Staroszczyk [105] notes that the currently proposed theoretical solutions for the mechanics of brittle ice [98], [106]–[108] are too complicated to be effectively implemented into practical

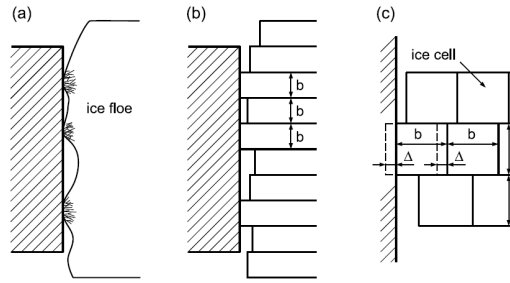


Figure 2.20: (a) Imperfect contact between an ice floe and a structure wall, (b) problem idealization, (c) detailed view and definitions. From [109].

engineering applications, and he suggests a different analysis of the mechanism of brittle failure of floating ice [109]. The analysis discusses a floe impacting a rigid cylindrical structure under several simplifying assumptions.

In this method, the floe (Figure 2.20) is treated as a collection of regular cells, each having a square shape with side b and thickness h . The dimension b is chosen to be close to h , if the geometry allows picking such values. As a particular cell starts to interact with the structure, it fails when the whole discrete element is advanced by a certain distance – critical displacement at which the crushing of ice occurs.

The model incorporates imperfectness of the floe–structure contact and the stochastic nature of the failure process. The failure stresses in ice are random variables to reproduce the large scatter in the empirical data. Simulating a large number of impacts produced distributions of the exerted forces. Three parameters describe the mechanism of fracture: (1) uniaxial compressive fracture stress, (2) uniaxial strain at which the fracture occurs, and (3) ice clearing stress. The latter is a stress occurring in already fractured blocks of ice. In this approach, many small-scale processes are disregarded. They are deemed unimportant because the purpose of

the analysis is to evaluate the total net forces sustained by the structure.

References

- [1] E. M. Schulson and P. Duval, *Creep and Fracture of Ice*. English. Leiden: Cambridge University Press, 2009, OCLC: 609840525, ISBN: 978-0-511-53983-1.
- [2] E. M. Schulson, “The structure and mechanical behavior of ice,” *Jom*, vol. 51, no. 2, pp. 21–27, 1999.
- [3] T. Sanderson, *Ice mechanics: risks to offshore structures*. London, UK ; Boston: Graham & Trotman, 1988, ISBN: 978-0-86010-785-9.
- [4] J. Glen, “The effect of hydrogen disorder on dislocation movement and plastic deformation of ice,” *Physik der kondensierten Materie*, vol. 7, no. 1, pp. 43–51, 1968.
- [5] C. Hayes and W. Webb, “Dislocations in ice,” *Science*, vol. 147, no. 3653, pp. 44–45, 1965.
- [6] P. Owston and K. Lonsdale, “The crystalline structure of ice,” *Journal of Glaciology*, vol. 1, no. 3, pp. 118–123, 1948.
- [7] M. Shokr and N. Sinha, *Sea ice: physics and remote sensing*, ser. Geophysical monograph 209. Washington, D.C. : Hoboken, New Jersey: American Geophysical Union ; John Wiley & Sons, 2015, OCLC: ocn889522621, ISBN: 978-1-119-02789-8.

- [8] B. Michel and R. Ramseier, "Classification of river and lake ice," *Canadian Geotechnical Journal*, vol. 8, no. 1, pp. 36–45, 1971.
- [9] A. Palmer, *Arctic offshore engineering*. World Scientific, 2013.
- [10] J. Schwarz and W. Weeks, "Engineering properties of sea ice," *Journal of Glaciology*, vol. 19, no. 81, pp. 499–531, 1977.
- [11] W. Weeks, *On Sea Ice*. University of Alaska Press, 2010, ISBN: 9781602231016.
- [12] J. P. Rafferty and Britannica Educational Publishing, *Glaciers, Sea Ice, and Ice Formation*. Chicago: Britannica Educational Pub., 2011, OCLC: 630537547, ISBN: 978-1-61530-189-8.
- [13] E. M. Schulson, "The fracture of ice ih," *Journal de Physique*, vol. 48, pp. 207–220, 1987.
- [14] V. F. Petrenko and R. W. Whitworth, *Physics of ice*, eng, 1. paperback issue; repr. 2006. Oxford: Oxford University Press, 2006, OCLC: 837076328, ISBN: 978-0-19-851894-5.
- [15] U. Nakaya, *Mechanical Properties of Single Crystals of Ice: Part 1: Geometry of Deformation*. US Army Snow Ice and Permafrost Research Establishment, Corps of Engineers, 1958.
- [16] R. L. Hooke, M. Mellor, W. Budd, J. Glen, A. Higashi, T. Jacka, S. Jones, R. Lile, R. Martin, M. Meier, *et al.*, "Mechanical properties of polycrystalline ice: An assessment of current knowledge and priorities for research: Report prepared for the international

- commission on snow and ice, with support from the us national science foundation,” *Cold regions science and technology*, vol. 3, no. 4, pp. 263–275, 1980.
- [17] J. Nadreau and B. Michel, “Ice properties in relation to ice forces,” *Second State-of-the-Art Report, Working Group on Ice Forces, IAHR Proceedings*, vol. 4, pp. 64–117, 1984.
 - [18] W. F. Weeks and A. J. Gow, “Preferred crystal orientations in the fast ice along the margins of the arctic ocean,” *Journal of Geophysical Research: Oceans*, vol. 83, no. C10, pp. 5105–5121, 1978.
 - [19] I. J. Jordaan, B. M. Stone, R. F. McKenna, and M. K. Fuglem, “Effect of microcracking on the deformation of ice,” *Canadian Geotechnical Journal*, vol. 29, no. 1, pp. 143–150, 1992.
 - [20] N. K. Sinha, P. Ehrhart, H. Carstanjen, A. Fattah, and J. Roberto, “Grain boundary sliding in polycrystalline materials,” *Philosophical Magazine A*, vol. 40, no. 6, pp. 825–842, 1979.
 - [21] N. Sinha, “Delayed elastic strain criterion for first cracks in ice,” Delft, The Netherlands, 1982.
 - [22] J. W. Glen, “The creep of polycrystalline ice,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 228, no. 1175, pp. 519–538, 1955.
 - [23] D. S. Sodhi and R. B. Haehnel, “Crushing ice forces on structures,” *Journal of cold regions engineering*, vol. 17, no. 4, pp. 153–170, 2003.

- [24] A. Palmer, D. Goodman, M. Ashby, A. Evans, J. Hutchinson, and A. Ponter, “Fracture and its role in determining ice forces on offshore structures,” *Annals of glaciology*, vol. 4, pp. 216–221, 1983.
- [25] Q. Yue, F. Guo, and T. Kärnä, “Dynamic ice forces of slender vertical structures due to ice crushing,” *Cold Regions Science and Technology*, vol. 56, no. 2-3, pp. 77–83, 2009.
- [26] G. W. Timco, “Indentation and Penetration of Edge-Loaded Freshwater Ice Sheets in the Brittle Range,” en, *Journal of Offshore Mechanics and Arctic Engineering*, vol. 109, no. 3, p. 287, 1987, ISSN: 08927219. DOI: 10.1115/1.3257022.
- [27] H. Peyton, “Sea ice forces. ice pressures against structures. national research council of canada,” Technical Memorandum, 92, 117-123, Tech. Rep., 1968.
- [28] K. Blenkarn *et al.*, “Measurement and analysis of ice forces on cook inlet structures,” in *Offshore Technology Conference*, Offshore Technology Conference, 1970.
- [29] Å. Ervik, T. S. Nord, K. V. Høyland, I. Samardzija, and H. Li, “Ice-ridge interactions with the norströmsgrund lighthouse: Global forces and interaction modes,” *Cold Regions Science and Technology*, vol. 158, pp. 195–220, 2019, ISSN: 0165-232X. DOI: <https://doi.org/10.1016/j.coldregions.2018.08.020>.
- [30] D. S. Sodhi, “Ice-induced vibrations of structures,” in *Proceedings of the Ninth International Association of Hydraulic Engineering and Research Symposium on Ice*, 1988, pp. 625–657.

- [31] Q. Yue, X. Zhang, X. Bi, and Z. Shi, “Measurements and analysis of ice induced steady state vibration,” in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2001.
- [32] L. Fransson and G. Danielsson, “Ice conditions and ice properties close to the light pier björnklacken shortly after failure due to forces from a moving ice sheet,” *Department of structural eng. Luleå University of Technology, Sweden, Skrift*, vol. 85, p. 13, 1985.
- [33] A. Palmer, Y. Qianjin, and G. Fengwei, “Ice-induced vibrations and scaling,” *Cold Regions Science and Technology*, vol. 60, no. 3, pp. 189–192, 2010, ISSN: 0165-232X. DOI: <https://doi.org/10.1016/j.coldregions.2009.11.005>.
- [34] F. Guo, Q. Yue, X. Bi, and Y. Liu, “Model test of ice-structure interaction,” in *ASME 2009 28th International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers, 2009, pp. 135–142.
- [35] D. S. Sodhi, “Ice-structure interaction during indentation tests,” in *Ice-structure interaction*, Springer, 1991, pp. 619–640.
- [36] D. S. Sodhi, “Crushing failure during ice–structure interaction,” *Engineering Fracture Mechanics*, vol. 68, no. 17-18, pp. 1889–1921, 2001.
- [37] Z. Kamio, T. Takawaki, H. Matsushita, T. Takeuchi, M. Sakai, T. Terashima, S. Akagawa, N. Nakazawa, H. Saeki, *et al.*, “Medium scale field indentation tests: Physical characteristics of first-year sea ice at notoro lagoon, hokkaido,” in *The Tenth International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2000.

- [38] M. Richard and R. S. Taylor, “Analysis of high pressure zone attributes from tactile pressure sensor field data,” in *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, Citeseer, 2014, V010T07A048–V010T07A048.
- [39] J. Schwarz, R. Frederking, V. Gavrillo, I. Petrov, K.-I. Hirayama, M. Mellor, P. Tryde, and K. Vaudrey, “Standardized testing methods for measuring mechanical properties of ice,” en, *Cold Regions Science and Technology*, vol. 4, no. 3, pp. 245–253, Jul. 1981, ISSN: 0165232X. DOI: 10.1016/0165-232X(81)90007-0.
- [40] D. Cole, “Effect of grain size on the internal fracturing of polycrystalline ice,” CRREL Report 86-5, Tech. Rep., 1986.
- [41] E. Schulson, “The brittle compressive fracture of ice,” *Acta Metallurgica et Materialia*, vol. 38, no. 10, pp. 1963–1976, Oct. 1990, ISSN: 09567151. DOI: 10.1016/0956-7151(90)90308-4.
- [42] S. J. Jones, R. Gagnon, A. Derradji, and A. Bugden, “Compressive strength of iceberg ice,” *Canadian Journal of Physics*, vol. 81, no. 1-2, pp. 191–200, 2003.
- [43] R. Lee and E. Schulson, “The Strength and Ductility of Ice Under Tension,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 110, no. 2, p. 187, 1988, ISSN: 08927219. DOI: 10.1115/1.3257049.
- [44] J. Lian, Q. Ouyang, X. Zhao, F. Liu, and C. Qi, “Uniaxial compressive strength and fracture mode of lake ice at moderate strain rates based on a digital speckle correlation method for deformation measurement,” *Applied Sciences*, vol. 7, no. 5, p. 495, 2017.

- [45] M. T. Boroojerdi, R. Taylor, S. Mohammadafzali, E. Bailey-Dudley, I. Turnbull, R. Hossain, *et al.*, “Field experiments on shear strength of solid and freeze-bonded sea ice,” in *OTC Arctic Technology Conference*, Offshore Technology Conference, 2018.
- [46] I. L. Meglis, P. M. Melanson, and I. Jordaan, “High speed testing of freshwater granular ice,” National Research Council Canada, Institute for Marine Dynamics, St. John’s, NL, Canada, Tech. Rep., 1998.
- [47] P. K. Dutta, D. M. Cole, E. M. Schulson, D. S. Sodhi, *et al.*, “A fracture study of ice under high strain rate loading,” *International Journal of Offshore and Polar Engineering*, vol. 14, no. 03, 2004.
- [48] H. Kim and J. N. Keune, “Compressive strength of ice at impact strain rates,” *Journal of materials science*, vol. 42, no. 8, pp. 2802–2806, 2007.
- [49] A. Combescure, Y. Chuzel-Marmot, and J. Fabis, “Experimental study of high-velocity impact and fracture of ice,” *International Journal of Solids and Structures*, vol. 48, no. 20, pp. 2779–2790, 2011.
- [50] S. I. Rogachko, G. N. Evdokimov, M. V. Melnikov, T. Kärnä, and E. Lehmus, “The influence of porosity on mechanical strength of hummocks,” in *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, 1997, pp. 151–158.
- [51] W. F. Smith and J. Hashemi, *Foundations of materials science and engineering*, 4th ed, ser. McGraw-Hill series in materials science. Boston: McGraw-Hill, 2005, ISBN: 978-0-07-295358-9.

- [52] J. Wells, I. Jordaan, A. Derradji-Aouat, and R. Taylor, “Small-scale laboratory experiments on the indentation failure of polycrystalline ice in compression: Main results and pressure distribution,” *Cold Regions Science and Technology*, vol. 65, no. 3, pp. 314–325, 2011.
- [53] K. B. Habib, R. S. Taylor, S. Bruneau, and I. J. Jordaan, “Experimental Study of Dynamics During Crushing of Freshwater Truncated Conical Ice Specimens,” ASME, May 2015, V008T07A014, ISBN: 978-0-7918-5656-7. DOI: 10.1115/OMAE2015-41904.
- [54] I. J. Jordaan and G. W. Timco, “Dynamics of the ice-crushing process,” *Journal of Glaciology*, vol. 34, no. 118, pp. 318–326, 1988.
- [55] T. Browne, R. Taylor, I. Jordaan, and A. Gurtner, “Small-scale ice indentation tests with variable structural compliance,” *Cold Regions Science and Technology*, vol. 88, pp. 2–9, Apr. 2013, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2012.12.006.
- [56] B. J. O’Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, “Experimental investigation of oscillation of loads in ice high-pressure zones, part 1: Single indenter system,” *Cold Regions Science and Technology*, vol. 124, pp. 25–39, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.005.
- [57] P. Birajdar, R. Taylor, K. Habib, and R. Hossain, “Analysis of Medium-Scale Laboratory Tests on Ice Crushing Dynamics,” Offshore Technology Conference, 2016. DOI: 10.4043/27482-MS.

- [58] P. Birajdar, R. Taylor, and R. Hossain, "Analysis of the Effect of Structural Compliance During Medium-Scale Laboratory Tests on Ice Crushing Dynamics," San Francisco, California, USA, Jun. 2017.
- [59] T. Brown, "Four years of ice force observations on the confederation bridge," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2001.
- [60] J. S. Tibbo, K. M. Obert, N. Shrestha, D. Tripathi, D. C. Mayne, and T. G. Brown, "Year twelve of the confederation bridge ice monitoring program," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2009.
- [61] J. Schwarz and P. Jochmann, "Ice forces affected by temperature and thickness of the ice," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2009.
- [62] R. Weiss, B. Wright, and B. Rogers, "In-ice performance of the molikpaq off sakhalin island," in *Proceedings of the 16th International Conference on Port and Ocean Engineering under Arctic Conditions POAC*, vol. 1, 2001.
- [63] S. Vershinin, P. Truskov, and K. Kuzmichev, "Sakhalin island–offshore platform structures–the impact and influence of ice," *Institute Giprostroymost, Moscow, Russia*, 2006.
- [64] M. Jefferies, "Dynamic response of" molikpaq" to ice-structure interaction.,” *Proceedings of the 7th OMAE, Houston, February 7-12, 1988*, vol. 4, pp. 201–220, 1988.

- [65] M. Jefferies, B. Rogers, M. Hardy, and B. Wright, "Ice load measurement on molik-paq: Methodology and accuracy," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2011.
- [66] G. Crocker, K. Croasdale, R. McKenna, G. English, J. Guzzwell, and S. Bruneau, "C-core iceberg impact experiment-phase 2, final report," *C-CORE, Memorial University of Newfoundland, Newfoundland*, 1997.
- [67] F. Ralph, R. McKenna, G. Crocker, and K. Croasdale, "Pressure/area measurements from the grappling island iceberg impact experiment," *Proc. IAHR, St. Petersburg*, 2004.
- [68] B. Wright, "Ice loads on the kulluk in managed ice conditions," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2001.
- [69] M. Metge, B. W. Danielewicz, and R. Hoare, *On Measuring Large Scale Ice Forces, Hans Island, 1980: Presentation to POAC, July 1981*. Pallister Resource Management Limited, 1987.
- [70] A. Murdza, A. Marchenko, A. Sakharov, P. Chistyakov, E. Karulin, and M. Karulina, "Test with L-shaped cantilever beam for complex shear and bending strength," in *Proceedings of the 23rd IAHR International Symposium on Ice.*, 2016.
- [71] D. Kheisin and N. Cherepanov, "Change of ice structure in the zone of impact of a solid body against the ice cover surface," *Problems of the Arctic and Antarctic*, vol. 33, p. 239, 1973.

- [72] R. Gagnon, “Analysis of visual data from medium scale indentation experiments at hobson’s choice ice island,” *Cold Regions Science and Technology*, vol. 28, no. 1, pp. 45–58, 1998.
- [73] C. C. Guard and C. Daley, *Compilation of Medium Scale Ice Indentation Test Results and Comparison to ASPPR*. National Research Council of Canada, 1994.
- [74] R. Johnson, J. Benoit, *et al.*, “Iceberg impact strength,” in *Offshore Technology Conference*, Offshore Technology Conference, 1987.
- [75] R. Meaney, I. Jordaan, and J. Xiao, “Analysis of medium scale ice-indentation tests,” *Cold Regions Science and Technology*, vol. 24, no. 3, pp. 279–287, 1996, ISSN: 0165-232X. DOI: [https://doi.org/10.1016/0165-232X\(95\)00025-7](https://doi.org/10.1016/0165-232X(95)00025-7).
- [76] H. Sacki, K.-i. Hiyayama, T. Kawasaki, S. Akagawa, K. Kato, K. Kamesaki, K. Saka, A. Kurokawa, *et al.*, “Joia project of study on ice load,” in *The Seventh International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 1997.
- [77] D. S. Sodhi, T. Takeuchi, M. Kawamura, N. Nakazawa, and S. Akagawa, “Measurement of ice forces and interfacial pressure during medium-scale indentation tests in japan,” in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2001.
- [78] R. Taylor, “Analysis of Scale Effect in Compressive Ice Failure and Implications for Design,” PhD thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, 2010.

- [79] R. Frederking, “Ice pressure variations during indentation,” in *Proceedings of the 17th IAHR International Symposium on Ice*, vol. 2, 2004, p. 307.
- [80] K. Kennedy, I. Jordaan, M. Maes, and A. Prodanovic, “Dynamic activity in medium-scale ice indentation tests,” *Cold regions science and technology*, vol. 22, no. 3, pp. 253–267, 1994.
- [81] P. Birajdar, “Analysis of Compressive Ice Failure and Ice Crushing Dynamics during Medium-Scale Indentation Tests,” Master’s thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, May 2017.
- [82] I. J. Jordaan, “Mechanics of ice-structure interaction,” *Engineering Fracture Mechanics*, vol. 68, no. 17-18, pp. 1923–1960, Dec. 2001, ISSN: 00137944. DOI: 10.1016/S0013-7944(01)00032-7.
- [83] M. Fuglem, K. Muggeridge, *et al.*, “Design load calculations for iceberg impacts,” *International Journal of Offshore and Polar Engineering*, vol. 9, no. 04, 1999.
- [84] S. K. Singh, I. J. Jordaan, J. Xiao, and P. A. Spencer, “The flow properties of crushed ice,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 117, no. 4, pp. 276–282, 1995.
- [85] I. J. Jordaan, M. A. Maes, P. W. Brown, and I. P. Hermans, “Probabilistic analysis of local ice pressures,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 115, no. 1, pp. 83–89, 1993.

- [86] I. J. Jordaan, D. G. Matskevitch, and I. L. Meglis, “Disintegration of ice under fast compressive loading,” *International Journal of Fracture*, vol. 97, no. 1, pp. 279–300, 1999, ISSN: 1573-2673. DOI: 10.1023/A:1018605517923.
- [87] J. Dempsey, A. Palmer, and D. Sodhi, “High pressure zone formation during compressive ice failure,” *Engineering fracture mechanics*, vol. 68, no. 17-18, pp. 1961–1974, 2001.
- [88] B. J. O’Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, “Experimental investigation of oscillation of loads in ice high-pressure zones, part 2: Double indenter system — Coupling and synchronization of high-pressure zones,” *Cold Regions Science and Technology*, vol. 124, pp. 11–24, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.002.
- [89] B. Spektor, “Antarctica’s Largest Iceberg Is About to Die ... Near the Equator,” Jun. 2018. [Online]. Available: <https://www.space.com/40826-largest-antarctic-iceberg-b15-is-melting.html> (visited on 08/12/2019).
- [90] P. Kry, “A statistical prediction of effective ice crushing stresses on wide structure.,” in *Proc. of IAHR Ice Symposium.*, 1978.
- [91] M. Ashby, A. Palmer, M. Thouless, D. Goodman, M. Howard, S. Hallam, S. Murrell, N. Jones, T. Sanderson, A. Ponter, *et al.*, “Nonsimultaneous failure and ice loads on arctic structures,” in *Offshore Technology Conference*, Offshore Technology Conference, 1986.

- [92] R. S. Taylor, I. J. Jordaan, C. Li, and D. Sudom, “Local design pressures for structures in ice: Analysis of full-scale data,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 132, no. 3, p. 031 502, 2010.
- [93] F. Ralph and I. Jordaan, “Probabilistic methodology for design of arctic ships,” in *ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers, 2013, V006T07A010–V006T07A010.
- [94] M. S. Rahman, R. S. Taylor, A. Kennedy, A. S. Ré, and B. Veitch, “Probabilistic analysis of local ice loads on a lifeboat measured in full-scale field trials,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 137, no. 4, p. 041 501, 2015.
- [95] A. Suyuthi, B. Leira, and K. Riska, “A generalized probabilistic model of ice load peaks on ship hulls in broken-ice fields,” *Cold Regions Science and Technology*, vol. 97, pp. 7–20, 2014.
- [96] R. S. Taylor and M. Richard, “Development of a probabilistic ice load model based on empirical descriptions of high pressure zone attributes,” in *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, Citeseer, 2014, V010T07A049–V010T07A049.
- [97] R. S. Taylor, M. Richard, and R. Hossain, “A probabilistic high-pressure zone model for local and global loads during ice-structure interactions,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 141, no. 5, p. 051 604, 2019.
- [98] M. Ashby and S. Hallam, “The failure of brittle solids containing small cracks under compressive stress states,” *Acta Metallurgica*, vol. 34, no. 3, pp. 497–510, 1986.

- [99] K. Kolari, “A complete three-dimensional continuum model of wing-crack growth in granular brittle solids,” *International Journal of Solids and Structures*, vol. 115, pp. 27–42, 2017.
- [100] M. Ashby and C. Sammis, “The damage mechanics of brittle solids in compression,” *Pure and Applied Geophysics*, vol. 133, no. 3, pp. 489–521, 1990.
- [101] V. G. Tsuprik *et al.*, “Theoretical and experimental studies of specific energy of mechanical failure of sea ice,” in *The Twenty-second International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2012.
- [102] A. T. Bekker, V. G. Tsuprik, E. E. Pomnikov, *et al.*, “Studies of specific energy fracture of ice using method test samples on uniaxial compression,” in *The 27th International Ocean and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2017.
- [103] V. Tsuprik, A. Bekker, E. Pomnikov, and E. Ivolgin, “Experimental researching of the specific energy mechanical fracture of ice by method of uniaxial compression of samples,” in *The 27th International Ocean and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2017.
- [104] I. Hawkes and M. Mellor, “Deformation and fracture of ice under uniaxial stress,” *Journal of Glaciology*, vol. 11, no. 61, pp. 103–131, 1972.
- [105] R. Staroszczyk, Staroszczyk, and Buettner, *Ice Mechanics for Geophysical and Civil Engineering Applications*. Springer, 2019.

- [106] S.-G. Sjölin, “A constitutive model for ice as a damaging visco-elastic material,” *Cold Regions Science and Technology*, vol. 14, no. 3, pp. 247–262, 1987.
- [107] W. A. Nixon, “Wing crack models of the brittle compressive failure of ice,” *Cold regions science and technology*, vol. 24, no. 1, pp. 41–55, 1996.
- [108] A. Pralong, K. Hutter, and M. Funk, “Anisotropic damage mechanics for viscoelastic ice,” *Continuum Mechanics and Thermodynamics*, vol. 17, no. 5, pp. 387–408, 2006.
- [109] R. Staroszczyk, “Loads on an off-shore structure due to an ice floe impact,” *Archives of Hydro-Engineering and Environmental Mechanics*, vol. 54, no. 2, pp. 77–94, 2007.

3 | Cohesive Zone Micromechanical Model for Compressive and Tensile Failure of Polycrystalline Ice

Preface

This chapter consists of the research work published in Engineering Fracture Mechanics¹. Note that the text of the chapter slightly differs from the published version due to the comments and clarifications addressed during thesis review. This project was in development for about a year, and I am the primary author of the modeling approach. My supervisors Dr. Robert Sarracino and Dr. Rocky Taylor, who are co-authors on this publication, provided valuable feedback about the physical properties of ice, various aspects of ice mechanics and the processes to be captured by a simulation.

After writing down the mathematical formulation, the model was implemented in C#, C++, and CUDA. The need to use several programming languages stems from the use of GPU acceleration and third-party libraries, including Intel MKL. A particular effort was made to visualize the damaged cohesive zones in a suitable form. I wrote the computer code and prepared the test setups, which are now freely distributed as open-source. I also performed

¹<https://doi.org/10.1016/j.engfracmech.2018.04.023>

the simulations and prepared the manuscript. Dr. Robert Sarracino and Dr. Rocky Taylor reviewed and revised the manuscript, contributed to the introduction and the conclusion, and helped with responding to the comments from the reviewers during the publication process.

Abstract

In this work, a cohesive zone model is applied to model fracture behavior of polycrystalline cylindrical samples under uniaxial loading conditions. The motivation of the research is the need to examine the approaches and methodologies that are promising for capturing both the continuum and the brittle behavior of ice. The model is based on the implicit finite element method that is combined with Park-Paulino-Roesler formulation for cohesive potential, which is a new approach for modeling ice. We implement an adaptive time stepping scheme that takes into account the rate of damage and failure of cohesive zones. Material properties and model parameters have been calibrated using available experimental data for laboratory-made freshwater ice samples. Simulations are performed for samples with different grain sizes, and the resulting stress-strain and damage accumulation curves are recorded. Investigation of the dependency between the grain size and fracture strength shows the grain-boundary strengthening effect (Hall-Petch dependence) that is consistent with experimental results. Fracture patterns observed in simulated failure events are also consistent with observations from experiments.

3.1 Introduction

Numerical modelling of ice failure is an ongoing research topic where one goal is to predict the failure behavior of naturally occurring ice under various loading conditions while accounting for statistical variability. Design codes presently used for ice-strengthened structures rely largely on methods derived from empirical relations obtained from the analysis of full-scale data (e.g. [1]). New physics-based models of ice failure that link interaction loads with the mechanics of ice failure processes would complement current design load methodology. Computer simulations that capture physical processes that occur during experiments offer the potential to allow study of scenarios not readily accessible to experimentation and to support the interpretation of full-scale ice load data. At the same time, ice is a very challenging material to simulate using numerical modelling due to the complex interplay between continuum and discrete failure processes that occur during ice compressive failure [2], [3]. In this work we investigate the stress-strain relationships of ice samples up to the point of failure for uniaxial specimens under compression and tension, as a means to investigate the modelling of accumulating microcracking damage and corresponding macroscopic fractures that ultimately cause specimen failure in unconfined specimens.

Among the various types of naturally occurring ice, glacial ice typically has a granular microstructure and is formed from fresh water (snow) with few impurities. To study the mechanical properties of freshwater ice in the laboratory, test specimens are typically made from equiaxed polycrystalline freshwater ice [4]. During full-scale interactions the state of stress is

typically triaxial because of high confinement[5]. Modelling the uniaxial compressive behavior of ice is an essential first step in the development of a numerical framework that can be expanded upon to account for more complex states of stress. For this reason, in the present work emphasis is placed on modelling failure behavior of ice under uniaxial conditions. Triaxial ice failure behavior will be considered in future work.

For validation purposes, measurements of uniaxial ice compressive strength are readily available and new experiments can be carried out using a standard materials testing apparatus placed in a sub-zero environmental chamber. From such experiments on polycrystalline ice samples, it is evident that repeat measurements will vary due to natural variability of the material, such as different sizes and arrangements of grains [6]. However, average trends in the strength behavior do indicate that grain size, temperature, and strain rate have significant influence on the measured compressive strength, making it possible to estimate the average uniaxial failure stress from material properties and loading conditions [5], [7].

Usually, models apply established techniques with various modifications to focus on certain aspects of ice behavior. For example, polycrystalline samples can be viewed as solids comprised of grains with accumulated microscopic damage. Such a concept is based on experimental observations of ice damage [8], [9]. The finite element technique (FE) is well-suited for modeling deformable solids, where the number of interacting fragments is limited. With regard to ice, this technique is suitable for processes that lead to moderate fragmentation, where individual pieces remain solid, but are not pulverized into fine crushed particles. In cases where geometries of individual grains are not important, or where extreme crushing or distortion has

taken place in sub domains of the larger region of interest, particle-based or meshless methods can be more computationally efficient. The current work focuses on fractures that originate between individual grains since grain boundaries are known to produce local stress concentrations in polycrystalline ice (e.g. [10]) and hence their precise shapes are kept as tetrahedral meshes and the finite element method is used.

A micromechanical model explicitly tracks microscopic damage associated with microcracking and local deformation for given macroscopic load states. It is a direct mathematical description of the underlying physical processes based on continuum mechanics. In the context of ice fracture, the applicability of this model is limited to certain types of behavior and a given range of parameters associated with brittle fracture. For example, the model works well in situations with a single or a few cracks and limited fragmentation, but performs poorly in crushing scenarios where the number of the resulting fragments is comparable to the number of grains in the sample. Future expansion of the model is planned to incorporate failure phenomena associated with other states of stress and more complex loading conditions.

In this paper, individual grains are treated as linear elastic solids within a FE framework, with an intrinsic zone (CZ) model used to represent the fracture mechanism. More complex microscopic damage is modeled through degradation and eventual breakdown of cohesive zones. Strictly speaking, ice is a viscoelastic material [7], [11], but for small deformations and sufficiently high strain rates undamaged polycrystalline ice can be treated as a linear elastic material. A collision response scheme is applied to prevent the separated fragments from interpenetrating after the fracture occurs. Ice behavior is limited primarily to brittle fracture at

temperatures below -10°C and strain rates between 10^{-3} and 10^{-1}s^{-1} , where viscous flow may be assumed to be negligible for unconfined specimens [5]. Inertia forces are accurately accounted in the simulation. The modeling results are compared with uniaxial tests on small-scale laboratory samples.

3.1.1 Related work

Micromechanical models apply to heterogeneous materials, such as composites and polycrystals by resolving the individual constituents of the material. Premachandran and Horii [12] proposed a finite element micromechanics-based model that is applicable only for a limited range of deformation rates, where creeping and microcracking are the only dominant mechanisms of deformation. Their model predicts crack density and stress-strain curves for a two-dimensional indentation of an ice sheet.

More recent studies by Taylor and Jordaan [13] and Moore et al. [14] build on previous work of Jordaan and co-workers [15]–[18]. They model the accumulated damage as a parameter recorded per element in a finite element model, which affects the viscoelastic properties of the material. Their calculations are performed in ABAQUS with an explicit integration scheme. In this model, individual fractures are not explicitly modeled, but rather the effects of discrete microcracking events on the "smoothed" continuum response is governed by modeled damage parameters. This approach focuses on constitutive behavior covering a wide range of confinement conditions that may be present during an ice-structure interaction, but these aspects are

beyond the scope of the present work. Ejection of the material, which is often observed experimentally, is handled via removal of damaged elements from the simulation once the damage reaches a certain threshold.

Our work follows along the lines of Wang and Li [19], who employ FE to model intergranular fracture of polycrystalline alumina. In their work, the 3D sample is treated as a deformable heterogeneous solid with cohesive zones inserted between all grains at the beginning of the simulation. The dynamic FE simulation is evaluated in ABAQUS for different samples containing up to 100 grains. Grains are unbreakable parts of the tetrahedral mesh, between which the cohesive zones are inserted. In ABAQUS, the formulation of the CZ traction-separation relationships is the so-called linear softening model, which is based on effective displacements [20]. This formulation has maximum separation, maximum traction, and the softening point separation as parameters. The model works quite well in many cases, i.e. it captures the effects of fracture and damage accumulation, but lacks flexibility due to its simplicity. Its drawbacks are discussed in detail in the review by Park and Paulino [21]. In particular, the formulation does not distinguish between tension and compression, or between tangential and normal loads, which creates a problem when modelling compressive failure.

A cohesive zone formulation that addresses these problems is published by Park, Paulino and Roesler [22]. They propose potential-based CZ laws with 8 parameters that separately define behavior in normal and tangential directions. This approach makes the PPR model more flexible in comparison with other CZ formulations. It can cover a wide range of materials from brittle to ductile, as well as mixed-mode fracture. In addition, the resulting traction functions

have continuous derivatives almost everywhere, which is important for implicit computation.

3.1.2 Our contribution

We present a computational tool that models accumulation of damage due to microcracking via cohesive zone method and predicts deformation behavior of laboratory-sized freshwater ice samples. The model is based on a FE approach similar to Wang and Li [19]. Our contribution is the application of the more flexible PPR cohesive zone formulation of Park, Paulino and Roesler [22], which allows one to reproduce some of the complex behavior seen in ice [23], [24]. The model also includes penalty-based collisions to reproduce scenarios where the sample is repeatedly fractured and fragmented after the initial failure.

In the work presented here we model the same number of grains as in typical laboratory experiments, or at least get close to that number. One of the samples used by Schulson [24], whose experiments we reproduce, contained approximately 4000 grains. Our computational tool is able to handle such numbers and perform up to 1000 time steps, which is required to model the fracture process. As a result, we can model the stress-strain behavior and damage accumulation as it would occur in an actual laboratory sample.

To lower the overall number of time steps we apply an adaptive scheme for time step selection, which covers four orders of magnitude. For simple elastic loading, a linear process, the simulation runs at large time steps and a convergent solution is found quickly. Once the process becomes nonlinear, i.e. when the material begins to fracture, time steps are automatically re-

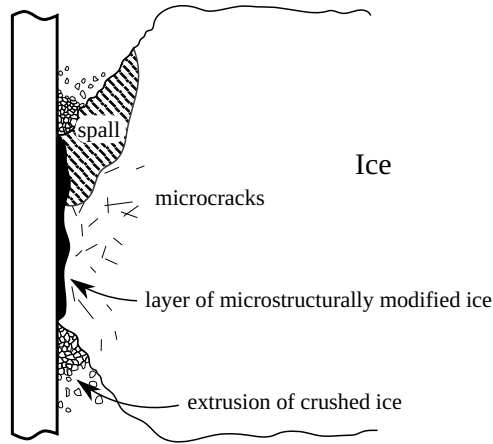


Figure 3.1: Schematic illustration of spalling, extrusion and high-pressure zone formation.

duced by the algorithm. With this scheme we are able to resolve crack propagation gradually, while keeping the total computational time under control. Additional details of the numerical implementation of this model may be found in work by Gribanov et al.²

3.2 Background

During interactions between ice and engineered structures, ice fracture mechanisms and associated loads can vary significantly for different geometries, temperatures and loading rates. Depending on the ice morphology, physical conditions and loading mechanism, ice can deform slowly in creep mode, fail in flexure, accumulate as rubble, and crush during an interaction [5]. Various types of experiments have been used to study fundamental properties of ice; in design practice, data from measurements of ice loads on structures are an essential consideration.

²I.Gribanov, R.Taylor, R.Sarracino, Parallel Implementation of Implicit Finite Element Model with Cohesive Zones and Collision Response using CUDA. International Journal for Numerical Methods in Engineering. Under review.

During a full-scale interaction (Fig.3.1), areas of low confinement near the outer edge of the interaction zone are prone to spalling fractures, which localize the contact into high pressure zones (*hpzs*) through which the majority of loads are transmitted. In these *hpzs*, the state of stress is multiaxial and in the highly confined central region of these zones, dynamic recrystallization and localized pressure melting processes contribute to pressure softening behavior that ultimately results in the extrusion of a highly viscous layer of damaged ice from the contact zone [25]. During an ice interaction event, many of these *hpzs* are continuously forming and being destroyed, such that ice load models can be developed based on statistical distributions of these *hpzs* acting over the interaction area [13]. While a number of recent experimental studies of *hpz* behavior during ice-structure interactions have been carried out at small-scale [26]–[30] and medium-scale [31], [32], these scenarios are beyond the scope of this work. As a starting point we study fracture in uniaxial laboratory experiments on freshwater ice, as this is seen as an essential first step in the development of a model which captures the discrete fracture processes that occur during such interactions.

Uniaxial tests measure compressive and tensile strength of cylindrical samples at a much smaller scale compared with the scales which apply to ice in nature. Schulson has conducted a series of these tests with granular freshwater ice at -10°C and strain rates of 10^{-3}s^{-1} [23], [24], which will be used to validate the model. At these strain rates ice behaves as a brittle solid; hence we will focus on brittle fracture. The length of the samples was 230 mm, and the diameter was 96 mm for compressive tests and 91 mm for tensile tests. Experimentally-obtained stress-strain curves show that failure stress decreases with grain size, although variability ex-

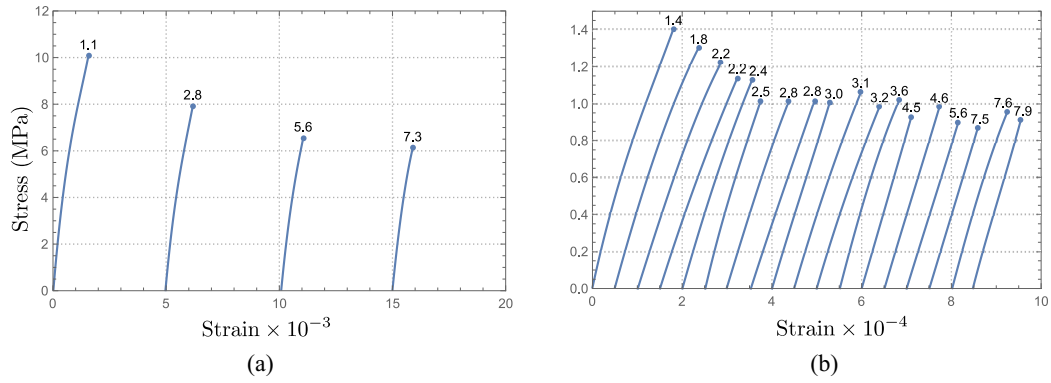


Figure 3.2: Stress-strain curves for granular freshwater ice Ih at -10°C , within the brittle regime at a strain rate of 10^{-3}s^{-1} , for specimens of different grain sizes. The curves are displaced along the abscissa for clarity. Grain sizes are shown in mm. Note the reduction in strength with increasing grain size. (a) Fracture in compression occurs after a strain of $\sim 10^{-3}$ and (b) in tension after $\sim 2 \times 10^{-4}$. Reproduced from work of Lee and Schulson.

ists between different regimes (Fig.3.2).

In compression tests where bonded caps were used, the failure occurred by shear faulting. The high-speed photography showed that the cracks had the same size as the grains and formed randomly within the sample, accumulating continuously until failure occurred. In compression, the fracture did not occur upon the formation of the first crack or even the first few cracks, but after the aggregate became heavily damaged [24].

In the tensile experiments the fracture surface was normal to the tensile axis and was highly faceted. The stress-strain curves were nonlinear, with a deviation of about 10% from a linear relationship. Absence of visible internal cracks was reported. The strength was 4-10 times lower than in compression, and the fracture occurred as soon as the first crack appeared. It was observed that the primary mode of failure was transgranular cleavage, at least for the specimens with the grain sizes over 4 mm [23].

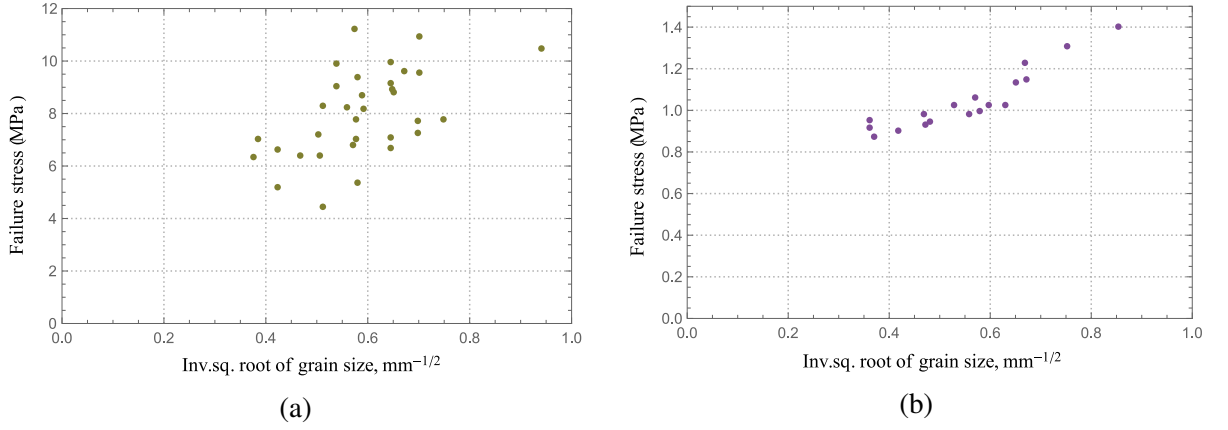


Figure 3.3: Fracture strength versus inverse square root of grain size for sample strained under (a) compression and (b) tension. The strain rate is 10^{-3} s^{-1} in both cases. Reproduced after Lee and Schulson.

With respect to grain size, the fracture behavior in both tension and compression followed the Hall-Petch equation [33]:

$$\sigma_y = \sigma_0 + \frac{k_y}{\sqrt{d}}, \quad (3.1)$$

where σ_y is the yield stress, σ_0 is a materials constant for the starting stress for dislocation movement (or the resistance of the lattice to dislocation motion), k_y is the strengthening coefficient (a constant specific to each material), and d is the average grain diameter. Fig.3.3 shows how experimental data follows this relationship.

In the proposed numerical model we attempt to reproduce the above observations without altering the model parameters between tensile and compressive tests or for changes in grain sizes. In other words, we run a number of simulations with constant properties of cohesive zones, then we compare the results to the experimental data shown above.

3.3 Simulation overview

Our model is based on a dynamic finite element method in which the deformable object consists of grains composed of multiple elements, each of which is discretized by a tetrahedral mesh (Fig.3.4). The grains are bonded to each other through cohesive zones, which have both shear and tensile resistance. Grains can separate from each other – through a numerical procedure of cohesive zone failure – but in the present model, individual grains cannot fracture, and their deformation is assumed to be linearly elastic. It is noted that future model implementations could account for transgranular fracture through incorporation of cohesive elements within individual grains, but such an approach has not been adopted here to reduce the computational cost (time required to perform calculations) and complexity of the model. At each time step, equations of motion are solved to determine subsequent mesh deformation. The Newmark-beta method is used for numerical integration, with time step selected by an adaptive scheme. The criteria for time step selection are convergence of Newton-Raphson iterations and the relative amount of damage accumulation per time step. Three contributing forces are considered, i.e. (1) elastic response, (2) traction forces from cohesive zones and (3) collision penalty forces. The formulation details for these forces are described in detail in the subsequent sections. An external load can be applied directly to the surface or through a colliding object.

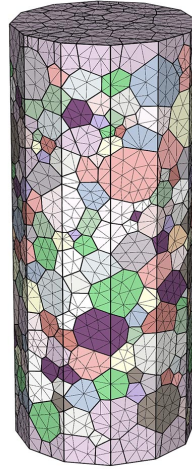


Figure 3.4: Cylinder with 1000 grains. Top and bottom caps do not contain cohesive zones.

3.3.1 Geometry of deformable samples

The polycrystalline geometry is generated with the open-source software Neper [34], which is a software package for polycrystal generation and meshing. It can deal with 2D and 3D polycrystals and has the following features:

- Generation of polycrystals from morphological properties, such as grain size distribution and grain shape distribution.
- Generation of multiscale microstructures. Each cell of a primary tessellation is partitioned into a new tessellation, and the process can be repeated any number of times.
- Generation of periodic or semi-periodic microstructures.
- Generation of uniform orientation distributions.
- Meshing with cohesive elements at interfaces.

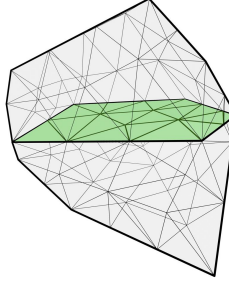


Figure 3.5: Two grains connected with cohesive zones.

- Visualization and analysis of the tessellations and meshes.

As mentioned above, deformable objects consist of grains. Each grain is partitioned with a tetrahedral mesh such that surfaces of adjacent grains have matching triangular meshes (Fig.3.5). Each tetrahedron composing the mesh is an element, and this procedure ensures that the facets connecting grains with their neighbors become triangular cohesive zones. In our simulations each grain was comprised of roughly 100 tetrahedral elements.

The ends of the cylindrical sample are single grains, which prevents the caps from fracturing. This configuration corresponds to bonded caps used in the experiments.

Typical grain size distribution is shown on Fig.3.6, where sizes are determined by applying the following transformation to grain volumes:

$$d_i = 2\sqrt[3]{v_i \frac{3}{4\pi}}, \quad (3.2)$$

where v_i is the volume of the polyhedron i , and d_i is the resulting grain size measure.

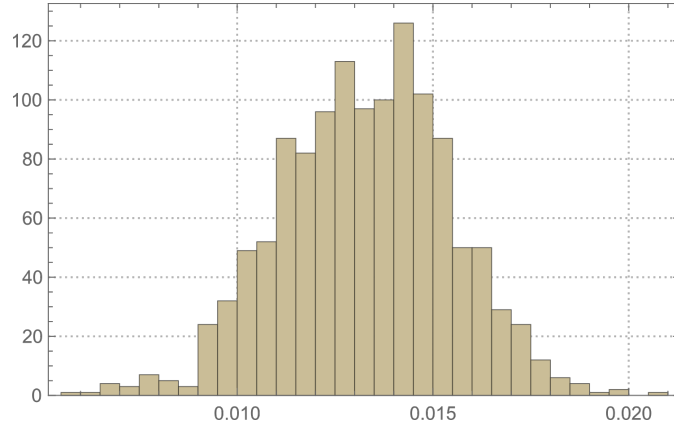


Figure 3.6: Distribution of grain sizes in a cylindrical sample with 1250 grains.

3.3.2 Integration and discretization schemes

Displacements of mesh nodes at time step n are represented by the vector

$$\mathbf{u}_n = [u_{x1}, u_{y1}, u_{z1}, \dots, u_{zN}]^T, \quad (3.3)$$

where N is the number of nodes that are free to move. Displacements at the subsequent time step are found from the non-linear vector equation

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} = \mathbf{f}_{ext} - (\mathbf{f}_{elem} + \mathbf{f}_{cz}) - \mathbf{D}\dot{\mathbf{u}}_{n+1}, \quad (3.4)$$

where \mathbf{M} is the mass matrix, \mathbf{D} is the Rayleigh damping matrix, \mathbf{f}_{ext} is the vector of external forces generated by the collisions, \mathbf{f}_{elem} represents internal elastic forces, and \mathbf{f}_{cz} is the vector of forces from the cohesive zones. Displacements and their derivatives at steps n and $(n+1)$ are related by Newmark-beta equations. The discretized equation of motion then becomes a

non-linear implicit equation

$$\mathbf{F}(\mathbf{u}_{n+1}) = \mathbf{0}, \quad (3.5)$$

which is resolved via Newton-Raphson iterations.

For some nodes the motion is prescribed by the boundary conditions. For example, when modeling compression of a cylindrical sample, the bottom of a cylinder is held in place, and the top is given downward velocity. Such "anchored" nodes have known displacements at each time step, therefore their coordinates are not part of the vector \mathbf{u}_{n+1} . Instead, they participate in equation (3.4) via force vectors \mathbf{f}_{elem} and \mathbf{f}_{cz} , which depend on the positions of all nodes, including the prescribed ones.

3.3.3 Elastic forces

For brittle ice behavior it is safe to assume that the deformation of the sample is small. Brittle failure leads to fragmentation, but the fragments themselves deform minimally. In this work we assume that the individual grains behave as linear elastic solids, which are free to deform, move and rotate. One suitable formulation of FE forces is given by Muller and Gross [35].

The forces on element nodes are represented by a 12-component vector

$$\mathbf{f}_{el} = [f_0^{(x)}, f_0^{(y)}, f_0^{(z)}, f_1^{(x)}, f_1^{(y)}, f_1^{(z)}, f_2^{(x)}, f_2^{(y)}, f_2^{(z)}, f_3^{(x)}, f_3^{(y)}, f_3^{(z)}]^T$$

In the following equation the \mathbf{x}_0 are the initial (material-space) nodal positions, and the \mathbf{x}_{n+1} are deformed (world-space) nodal positions at time step $(n+1)$. The force acting on the nodes is then given by

$$\mathbf{f}_{el} = \mathbf{R}\mathbf{K}(\mathbf{R}^{-1}\mathbf{x}_{n+1} - \mathbf{x}_0), \quad (3.6)$$

where \mathbf{K} is the constant stiffness matrix and \mathbf{R} is the 12×12 rotation matrix of the element.

3.3.4 Cohesive zone model

Cohesive zones are given significantly higher stiffness than the tetrahedral elements. This is to guarantee that the relative deformation of cohesive zones will be much less than the relative deformation of the grains, so that their effect on the volume of the object will be negligible.

Cohesive zone forces are computed as described by Park and Paulino [36]. The traction vector is found as the gradient of the potential function, which is

$$\begin{aligned} \Psi(\Delta_n, \Delta_t) = & \min(\phi_n, \phi_t) + \\ & \left[\Gamma_n \left(1 - \frac{\Delta_n}{\delta_n} \right)^\alpha \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^m + \langle \phi_n - \phi_t \rangle \right] \times \\ & \left[\Gamma_t \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^\beta \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^n + \langle \phi_t - \phi_n \rangle \right], \end{aligned} \quad (3.7)$$

Table 3.1: Properties of the material.

E	Young's modulus
ν	Poisson's ratio
ϕ_n, ϕ_t	Mode I and II fracture energies
σ_{max}	Normal cohesive strength
τ_{max}	Tangential cohesive strength
ρ	Density of the material

Table 3.2: Model parameters.

α, β	Shape of traction-separation curves
λ_n, λ_t	Non-dimensional slope indicators in PPR model
$\dot{\epsilon}$	Strain rate for load in uniaxial test
Δt	Initial time step

where $\langle \cdot \rangle$ is the Macaulay bracket, i.e.

$$\langle x \rangle = \begin{cases} 0 & x < 0, \\ x & x \geq 0. \end{cases} \quad (3.8)$$

PPR parameters $(\phi_n, \phi_t; m, n; \delta_n, \delta_t; \alpha, \beta)$ determine the properties of the material, such as fracture energy, cohesive strength and brittleness. Four of these parameters correspond to properties in the tangential direction, and four in the normal direction. Nomenclature is shown in Tables 3.1 and 3.2.

Energy constants Γ_n and Γ_t are defined by the expressions

$$\begin{aligned}\Gamma_n &= (-\phi_n)^{\langle \phi_n - \phi_t \rangle / (\phi_n - \phi_t)} \left(\frac{\alpha}{m} \right)^m, \\ \Gamma_t &= (-\phi_t)^{\langle \phi_t - \phi_n \rangle / (\phi_t - \phi_n)} \left(\frac{\beta}{n} \right)^n.\end{aligned}\tag{3.9}$$

when $\phi_n \neq \phi_t$, and

$$\Gamma_n = -\phi_n \left(\frac{\alpha}{m} \right)^m, \quad \Gamma_t = \left(\frac{\beta}{n} \right)^n\tag{3.10}$$

for the case when $\phi_n = \phi_t$.

Non-dimensional exponents m and n are related to initial slope indicators λ_n and λ_t by the following expressions

$$m = \frac{\alpha(\alpha - 1)\lambda_n^2}{(1 - \alpha\lambda_n^2)}, \quad n = \frac{\beta(\beta - 1)\lambda_t^2}{(1 - \beta\lambda_t^2)}.\tag{3.11}$$

The resulting traction-separation curves are continuous and differentiable. Negative separation for normal opening corresponds to compression of the connected elements, and the resulting negative force opposes interpenetration of the elements. The tangential relationship is an odd function, since the direction of tangential opening can be arbitrary (Fig.3.7). Maximum openings δ_n, δ_t can be expressed as functions of cohesive strengths [22], and the set of PPR parameters becomes $(\phi_n, \phi_t; m, n; \sigma_{max}, \tau_{max}; \alpha, \beta)$.

In order to obtain the traction force over the whole CZ, three-point Gaussian quadrature is applied. The integration points correspond to barycentric coordinates $(\frac{2}{3}, \frac{1}{6}, \frac{1}{6})$ and their permutations. The resulting forces are then distributed among the six nodes that define the CZ

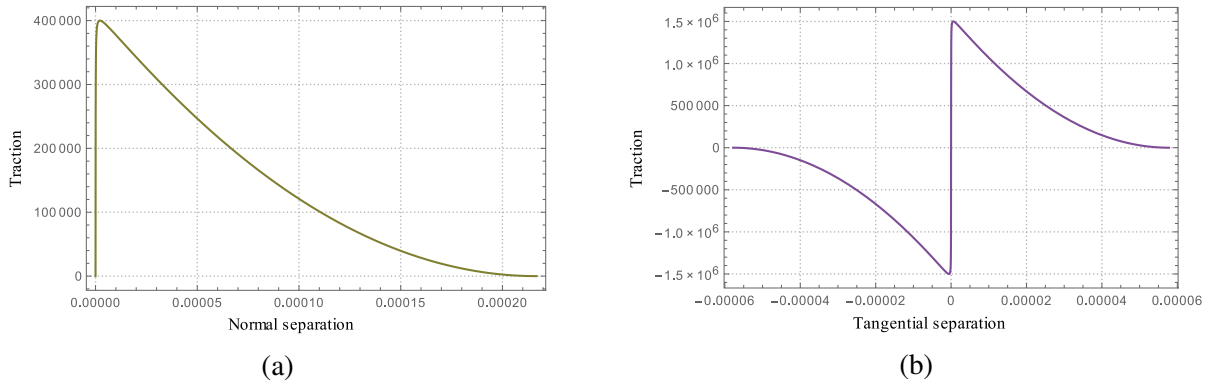


Figure 3.7: Traction-separation relations for (a) normal opening, and (b) tangential opening with parameter values ($\phi_n = \phi_t = 30$, $\sigma_{max} = 4 \times 10^5$, $\tau_{max} = 1.5 \times 10^6$, $\alpha = \beta = 3$, $\lambda_n = \lambda_t = 0.01$). Units are not included in this figure, because they can be selected later. Parameter values are selected somewhat arbitrarily, for illustration purpose.

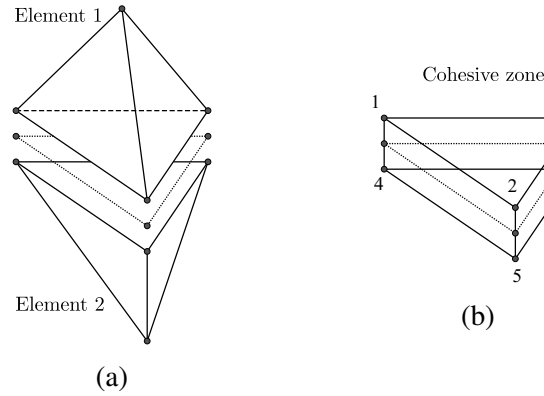


Figure 3.8: (a) Two tetrahedral elements with midplane and (b) corresponding cohesive zone formed by 6 nodes to which the resulting traction force is applied.

(Fig.3.8).

If a CZ is loaded in compression, a negative normal separation results in a negative traction force, which opposes interpenetration of the elements. The failure of a CZ represents the formation of a crack between the two adjacent grains. When CZ fails, it is removed from the simulation, and two free surfaces are created in its place. In order to properly account for interaction between the separated fragments, surface forces are explicitly added.

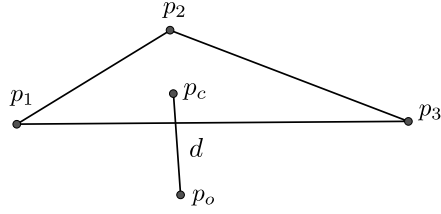


Figure 3.9: Geometry involved in collision response. The colliding node p_o is located inside the object close to the surface triangle (p_1, p_2, p_3) . The closest point on the surface is p_c . The distance between the colliding node and the surface is d .

3.3.5 Collision response

To the contacting surfaces of all objects we apply a penalty-based collision response, so that the surface forces appear alongside the other forces in the resulting equation of motion. Collisions can occur between fragments of a deformable object as well as between an object and a rigid body, such as an indenter. To avoid sudden jumps in forces when CZs fail, we also apply collision forces to all grains that are in contact, even when their cohesive zones are intact.

Contact forces are applied to all nodes that are inside of other tetrahedral elements. When a force is applied to a certain node, a counterforce is applied to the nearest surface triangle.

The force vector is computed as the gradient of the potential

$$\Psi = \frac{kd^2}{2}, \quad (3.12)$$

where d is the distance between the node and the surface triangle, and k is the stiffness constant (Fig.3.9). Such formulation results in nearly linear behavior for small penetration distances, requiring fewer Newton-Raphson iterations to converge and allowing for larger time steps. We

have found this relatively simple procedure adequate for modeling compressive tests, including both uniaxial compression and impact crushing.

3.3.6 Time step selection

The non-linearity of the processes requires continual monitoring to determine whether the time step needs to be reduced or whether it may safely be increased. We specifically monitor convergence of the Newton-Raphson iterations and accumulation of damage per simulation step. The time step is adjusted automatically when threshold values are exceeded. This procedure allows resolution of crack propagation when it develops, while running at larger steps when no damage occurs. Collisions between fast-moving fragmented material will cause reduction of time step as well, if no convergent solution is found. For static contacts, collision forces do not affect the time step, as these forces are nearly linear. The method for computing

a single simulation step is shown in Algorithm 1.

Data: current time step

Result: selection of the next time step

```
while  $iteration \leq maximum\ iterations$  do  
    assemble and solve the linear system;  
    if  $iteration > minimum\ iterations \wedge damage\ threshold\ exceeded$  then  
        discard calculation and reduce time step;  
        return  
    else if  $iteration > minimum\ iterations \wedge convergent\ solution\ found$  then  
        accept calculation; after  $N$  successful acceptances, increase time step for the next  
        cycle;  
    return
```

end

discard calculation and reduce time step;

return

Algorithm 1: Time step adjustment during a single simulation cycle. A result is either accepted or discarded, depending on the convergence of the solution and the proportion of damaged cohesive zones.

3.4 Numerical results

Tensile and compressive simulations were conducted for samples with 250, 500, 750, 1000, 1250, 1500 and 4000 grains. For each grain size, 1-10 samples were generated with different

Table 3.3: The modeled samples with different grain arrangements, i.e. different randomly generated geometries. The 4000-grain sample was only simulated in compressive test.

Number of grains	250	500	750	1000	1250	1500	4000
Number of samples	9	10	10	2	4	2	1

arrangements of grains (Table 3.3). Each simulation ran 1000-1600 steps and was stopped when the sample failed. Material properties and model parameters (Tables 3.4 and 3.5) were selected by matching the experimental data as closely as possible for the 250-grain samples (grain size about 20 mm).

Young's Modulus E was set from the initial slope of the stress-strain curves in Fig.3.2. The value of normal cohesive strength σ_{max} roughly corresponds to the maximum tensile stress the material can sustain, although tangential strength and grain geometry also influence the resulting macroscopic behavior. By comparing the simulation results of tensile and compressive tests, the tangential strength was set 3.75 times higher than the normal strength, which admittedly seems a bit high. The ratio determines the macroscopic behavior of the resulting material, i.e. its shear strength. A difference between tangential and normal strength is common in nature [37]. For most rocks the ratio is around 2.00, but no definitive ratio has been identified for ice. The comparison with the ratio for rocks is made because rock is a brittle material, whose fracture behavior in some cases is similar to that of the ice.

Cohesive zone shape parameters $\alpha, \beta, \lambda_n, \lambda_t$ were selected to match the brittle behavior of the material [22]. The initial time step Δt was selected so that the solution would not have convergence issues in the initial (linear) portion of the loading curve. Prescribed displacements

Table 3.4: Values of material properties.

E	10 GPa
ν	0.3
ϕ_n, ϕ_t	30 J m^{-2}
σ_{max}	$4 \times 10^5 \text{ Pa}$
τ_{max}	$1.5 \times 10^6 \text{ Pa}$
ρ	916.2 kg/m^3

Table 3.5: Values of simulation parameters.

α, β	3
λ_n, λ_t	0.01
$\dot{\epsilon}$	10^{-3} s^{-1}
Δt	0.005 s

were applied to the top layer of nodes of the cylinder to give it the constant strain rate $\dot{\epsilon}$. The bottom layer of nodes was fixed in place.

A typical resulting stress-strain curve is shown in Fig.3.10. The simulation process naturally separates into three stages: elastic deformation, damage accumulation and failure. During the first two stages, simulation time steps are relatively large and stay within the same order of magnitude as the initial time step. At the time of failure, the time step is automatically reduced by a factor of $\sim 10^4$ to resolve post-failure behavior and crack propagation. For the process depicted in Fig.3.10, the total simulated time was 0.5522 s, whereas the failure occurred in only $3 \times 10^{-5} \text{ s}$. By failure we denote the formation of the crack, which leads to subsequent decrease in loading force.

Crack propagation and damage accumulation for the compressive test are illustrated in Figs.

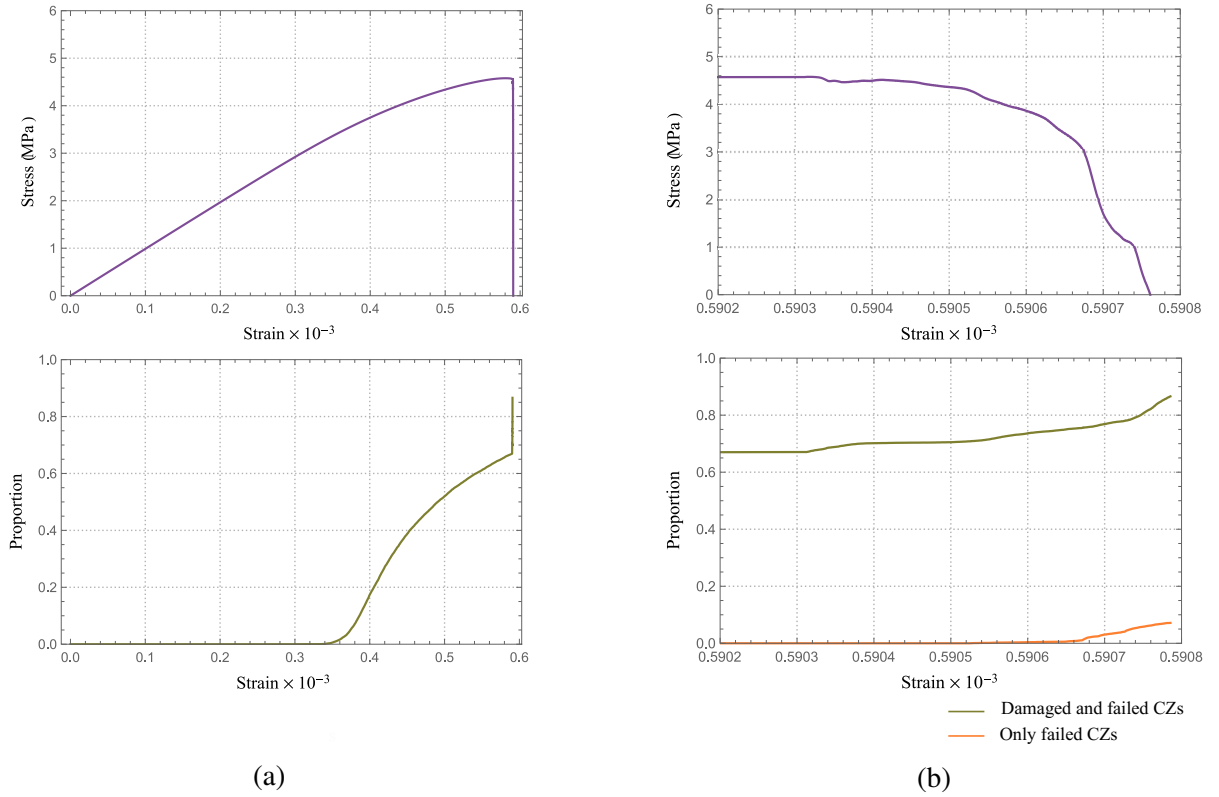


Figure 3.10: Stress-strain curves (top) and damage accumulation (bottom) for cylindrical sample with 500 grains loaded to failure in compression. Engineering stress at the top of the cylinder is used as a stress measure. (a) complete process showing 1365 computational steps. (b) the last 750 steps.



Figure 3.11: Accumulation of damaged cohesive zones in compressive test in a cylinder containing 1000 grains. Only pre-fail stages are shown, where the crack did not yet develop.

3.11 and 3.12. The damaged cohesive zones are oriented diagonally, which shows that the damage occurs in shear. At the point of failure, a characteristic brittle shear fault develops. These results are consistent with the experimental data, where similar diagonal damage and the characteristic shear fault were observed.

In tensile tests the stress-strain curves are similar, but the amount of accumulated damage is smaller, and the fracture strength is lower. Fig.3.13 shows the typical accumulation of damage during the simulation of a tensile test. Note that most damaged CZs are oriented horizontally, and the total amount of damage is smaller than in compressive tests. This is also consistent with experimental data. Horizontal orientation of the damaged CZs is explained by the uniaxial state of stress in the sample.

The resulting compressive strength of the samples can be plotted against the grain size simi-

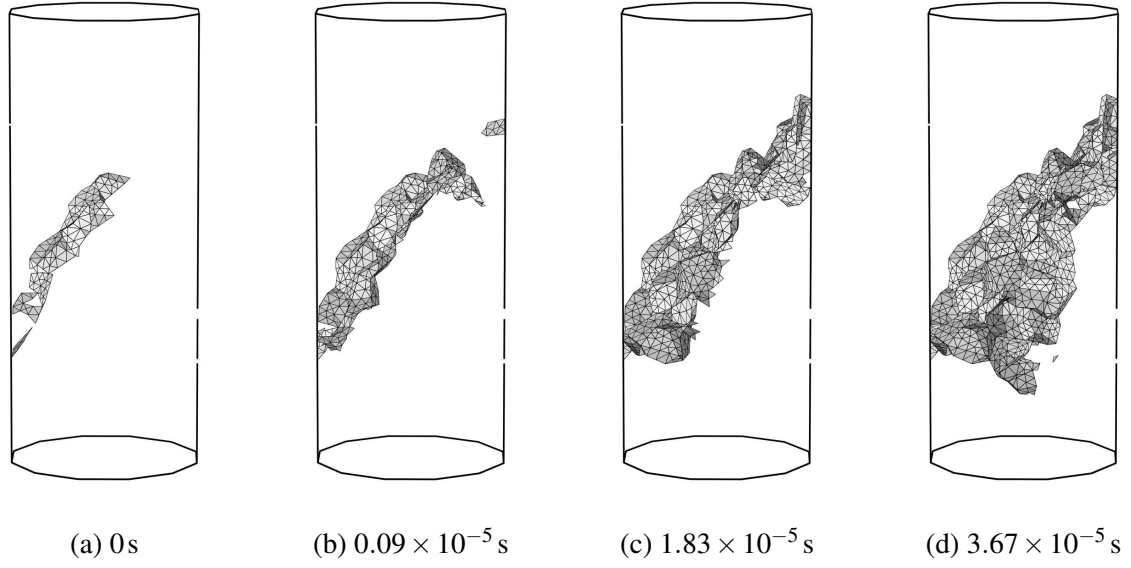


Figure 3.12: Propagation of crack in compressive test of a 1000-grain cylinder. Damaged cohesive zones are present in the model during the simulation, but are not shown on this figure for clarity. Time values are shown as offsets from $t = 0.638336$ s.

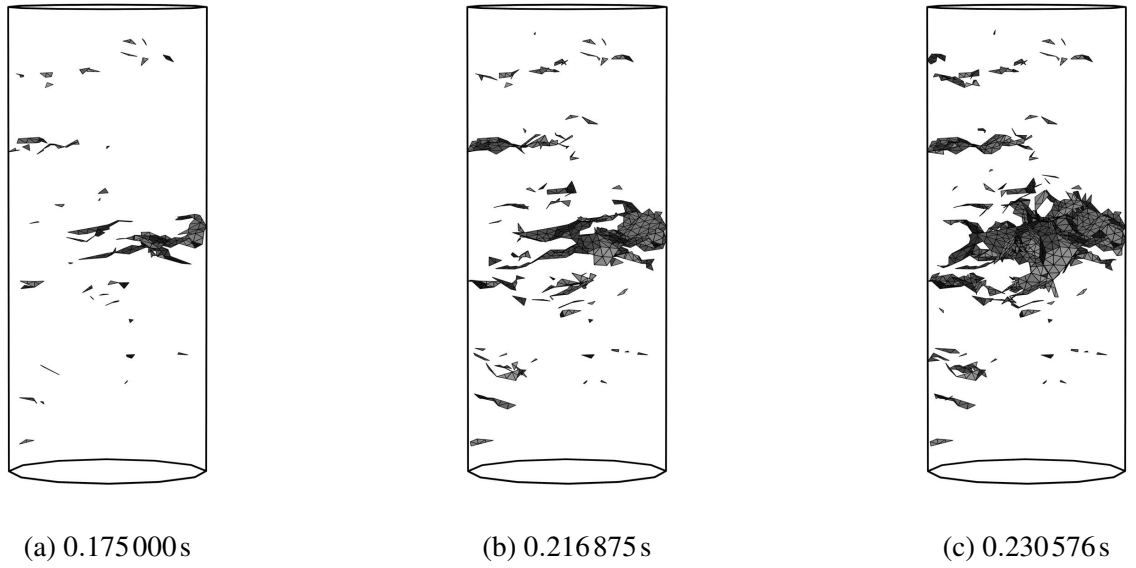


Figure 3.13: Damaged cohesive zones in tensile tests of a sample containing 1250 grains.

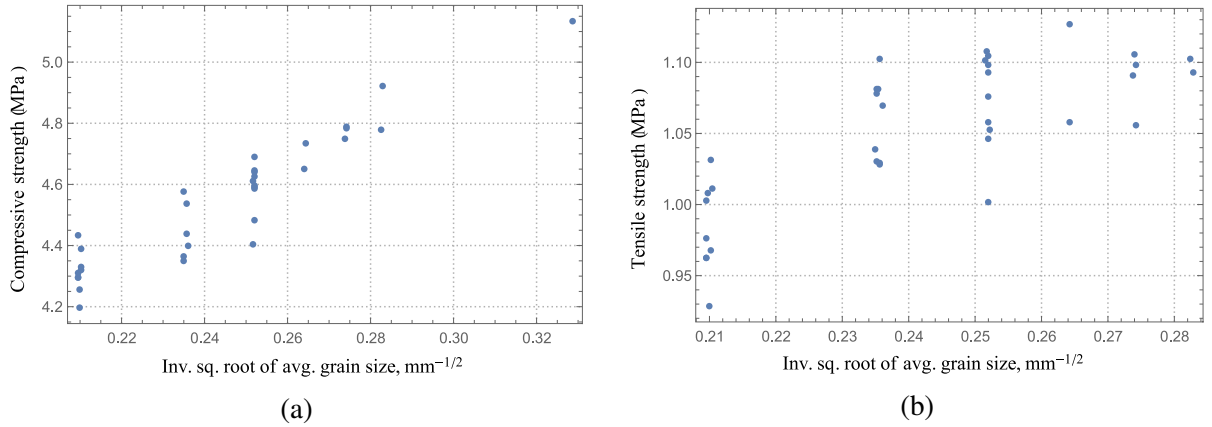


Figure 3.14: Modeled fracture strength versus inverse square root of grain size for tests in (a) compression and (b) tension.

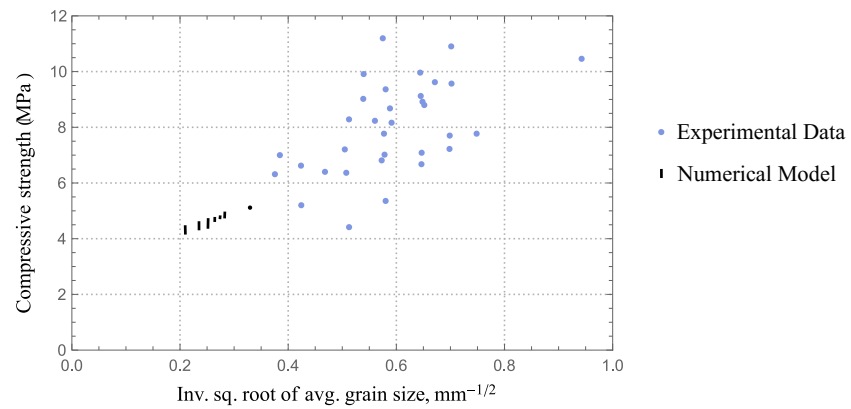


Figure 3.15: Comparison of the simulated compressive tests (Fig.3.14a) with the experimental data (Fig.3.3a).

larly to Fig.3.3. If the model reproduces the effect of grain boundary strengthening, then the obtained points will satisfy the Hall-Petch equation. As seen from Fig.3.14, this relationship is satisfied for compressive simulations, even though the model does not include crystal plasticity. For tensile tests, the points do not exactly follow the Hall-Petch strengthening, but the increasing trend exists. There may be several reasons why it is not followed precisely in tension simulation. (1) The shapes of the simulated grains are far more uniform than those found in nature, which affects their bonding strength. (2) Grain boundary strengthening only applies for a certain range of grain sizes – the material cannot be made infinitely strong by reducing the grains. (3) The proposed model does not account for crystal plasticity and intragranular fracture.

We can also compare the variability between the simulated samples and the experimental data. Both sets of data are shown on Fig.3.15 side by side. The simulated tests show less variability in compressive strength. The difference between the simulation and the experiments can be attributed to the difference in variability of grain sizes and cohesion between the grains. In simulations, all grains have similar size and volume, and parameters of cohesive zones are identical, while experimental samples show higher spread in grain sizes and fracture energies. Future work will examine how compound strength variability observed in experimental data can be better captured through modeling of variability in cohesive zone properties.

3.5 Conclusions and future work

In this work we have applied a micromechanical model to a whole specimen rather than to a representative volume. We reproduced uniaxial compression and tension tests in granular ice samples and were able to capture damage accumulation, crack propagation and even grain boundary strengthening. The model stands well against experimental validation. Notably, we found that the model, whose parameters were chosen by matching experimental results for samples of one particular grain size, successfully predicted the general trend of material strengthening with decreasing grain size. Extension of this work for samples with other grain shapes or for states of stress will be considered in the future. Computer code used for this computation is available as open source.³

As noted in Section 3.4, the grain geometry used in the work does not fully reflect the complexity of natural samples, since real grains show more variability in shapes and sizes. Similarly, the uniform strength assigned to cohesive zones in the simulations does not reflect the natural variability in grain-grain cohesion. Examination of the effects of introducing distributions in grain size and cohesive zone strengths is an area recommended for further study.

Our model treats individual grains as linear isotropic material, whereas ice crystals exhibit non-elastic and anisotropic behavior [7]. This simplification would affect the predicted stress-strain curves and failure stresses, but the exact magnitude of the effect is unknown without detailed study. The model accounts for neither transgranular fracture nor for viscous flow of

³<https://github.com/igorg520b/icFlow>

ice – phenomena whose effects are prominent in some scenarios, especially for lower strain rates, high confinement, or temperatures very near the melting point. In this work we only considered one select value for the strain rate, whereas rate sensitivity was demonstrated in laboratory experiments. These areas are recommended for further development.

Certain conditions in ice can lead to pressure softening, recrystallization and formation of high-pressure zones. In order to model these phenomena in the finite element framework, extension of the current model is needed. Our goal is to address these effects in our future work.

References

- [1] “Petroleum and natural gas industries – Arctic offshore structures,” International Organization for Standardization, Geneva, CH, Standard ISO 19906:2010(en), Dec. 2010.
- [2] R. Taylor, “Analysis of Scale Effect in Compressive Ice Failure and Implications for Design,” PhD thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, 2010.
- [3] R. Taylor and I. Jordaan, “Probabilistic fracture mechanics analysis of spalling during edge indentation in ice,” *Engineering Fracture Mechanics*, vol. 134, pp. 242–266, Jan. 2015, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2014.10.021.

- [4] D. M. Cole, "Preparation of polycrystalline ice specimens for laboratory experiments," *Cold Regions Science and Technology*, vol. 1, no. 2, pp. 153–159, Nov. 1979, ISSN: 0165232X. DOI: 10.1016/0165-232X(79)90007-7.
- [5] T. Sanderson, *Ice mechanics: risks to offshore structures*. London, UK ; Boston: Graham & Trotman, 1988, ISBN: 978-0-86010-785-9.
- [6] D. Cole, "Effect of grain size on the internal fracturing of polycrystalline ice," CRREL Report 86-5, Tech. Rep., 1986.
- [7] E. M. Schulson and P. Duval, *Creep and Fracture of Ice*. English. Leiden: Cambridge University Press, 2009, OCLC: 609840525, ISBN: 978-0-511-53983-1.
- [8] I. Meglis, P. Melanson, and I. Jordaan, "Microstructural change in ice: II. Creep behavior under triaxial stress conditions," *Journal of Glaciology*, vol. 45, no. 151, pp. 438–448, 1999, ISSN: 0022-1430, 1727-5652. DOI: 10.1017/S0022143000001295.
- [9] S. A. Snyder, E. M. Schulson, and C. E. Renshaw, "Effects of prestrain on the ductile-to-brittle transition of ice," *Acta Materialia*, vol. 108, pp. 110–127, Apr. 2016, ISSN: 13596454. DOI: 10.1016/j.actamat.2016.01.062.
- [10] R. Raj and M. Ashby, "On grain boundary sliding and diffusional creep," *Metallurgical Transactions*, vol. 2, no. 4, pp. 1113–1127, Apr. 1971, ISSN: 0360-2133, 1543-1916. DOI: 10.1007/BF02664244.
- [11] N. Sinha, P. Ehrhart, H. Carstanjen, A. Fattah, and J. Roberto, "Grain boundary sliding in polycrystalline materials," *Philosophical Magazine A*, vol. 40, no. 6, pp. 825–842, Dec. 1979, ISSN: 0141-8610, 1460-6992. DOI: 10.1080/01418617908234877.

- [12] R. Premachandran and H. Horii, “A Micromechanics-Based Constitutive Model for Polycrystalline Ice,” *Journal of Engineering Materials and Technology*, vol. 116, no. 3, p. 392, 1994, ISSN: 00944289. DOI: 10.1115/1.2904303.
- [13] R. S. Taylor and M. Richard, “Development of a Probabilistic Ice Load Model based on Empirical Descriptions of High Pressure Zone Attributes,” in *Proceedings of OMAE 2014*, 2014.
- [14] P. Moore, I. Jordaan, and R. Taylor, “Explicit Finite Element Analysis Of Compressive Ice Failure Using Damage Mechanics,” 2013.
- [15] I. J. Jordaan, D. G. Matskevitch, and I. L. Meglis, “Disintegration of ice under fast compressive loading,” *International Journal of Fracture*, vol. 97, no. 1, pp. 279–300, 1999, ISSN: 1573-2673. DOI: 10.1023/A:1018605517923.
- [16] C. Li, “Finite element analysis of ice-structure interaction with a viscoelastic model coupled with damage mechanics,” PhD thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, 2003.
- [17] J. Xiao, “Finite element modeling of damage processes in ice-structure interaction,” PhD thesis, Memorial University of Newfoundland, St. John’s, NL, Canada, 1991.
- [18] J. Xiao, “Damage and fracture of brittle viscoelastic solids with application to ice load models,” Ph.D. Memorial University of Newfoundland, St. John’s, NL, Canada, 1997.
- [19] Z. Wang and P. Li, “Voronoi cell finite element modelling of the intergranular fracture mechanism in polycrystalline alumina,” *Ceramics International*, vol. 43, no. 9, pp. 6967–6975, Jun. 2017, ISSN: 02728842. DOI: 10.1016/j.ceramint.2017.02.121.

- [20] P. H. Geubelle and J. S. Baylor, "Impact-induced delamination of composites: A 2d simulation," *Composites Part B: Engineering*, vol. 29, no. 5, pp. 589–602, Sep. 1998, ISSN: 13598368. DOI: 10.1016/S1359-8368(98)00013-4.
- [21] K. Park and G. Paulino, "Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces," en, *Applied Mechanics Reviews*, vol. 64, no. 6, p. 060 802, Feb. 2013, ISSN: 0003-6900. DOI: 10.1115/1.4023110.
- [22] K. Park, G. H. Paulino, and J. R. Roesler, "A unified potential-based cohesive model of mixed-mode fracture," *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 6, pp. 891–908, Jun. 2009, ISSN: 0022-5096. DOI: 10.1016/j.jmps.2008.10.003.
- [23] R. Lee and E. Schulson, "The Strength and Ductility of Ice Under Tension," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 110, no. 2, p. 187, 1988, ISSN: 08927219. DOI: 10.1115/1.3257049.
- [24] E. Schulson, "The brittle compressive fracture of ice," *Acta Metallurgica et Materialia*, vol. 38, no. 10, pp. 1963–1976, Oct. 1990, ISSN: 09567151. DOI: 10.1016/0956-7151(90)90308-4.
- [25] I. J. Jordaan, "Mechanics of ice-structure interaction," *Engineering Fracture Mechanics*, vol. 68, no. 17-18, pp. 1923–1960, Dec. 2001, ISSN: 00137944. DOI: 10.1016/S0013-7944(01)00032-7.
- [26] T. Browne, R. Taylor, I. Jordaan, and A. Gurtner, "Small-scale ice indentation tests with variable structural compliance," *Cold Regions Science and Technology*, vol. 88, pp. 2–9, Apr. 2013, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2012.12.006.

- [27] K. B. Habib, R. S. Taylor, S. Bruneau, and I. J. Jordaan, “Experimental Study of Dynamics During Crushing of Freshwater Truncated Conical Ice Specimens,” ASME, May 2015, V008T07A014, ISBN: 978-0-7918-5656-7. DOI: 10.1115/OMAE2015-41904.
- [28] B. J. O’Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, “Experimental investigation of oscillation of loads in ice high-pressure zones, part 1: Single indenter system,” *Cold Regions Science and Technology*, vol. 124, pp. 25–39, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.005.
- [29] B. J. O’Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, “Experimental investigation of oscillation of loads in ice high-pressure zones, part 2: Double indenter system — Coupling and synchronization of high-pressure zones,” *Cold Regions Science and Technology*, vol. 124, pp. 11–24, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.002.
- [30] R. Taylor, T. Browne, I. Jordaan, and A. Gurtner, “Fracture and Damage During Dynamic Interactions Between Ice and Compliant Structures at Laboratory Scale,” ASME, Jun. 2013, V006T07A024, ISBN: 978-0-7918-5540-9. DOI: 10.1115/OMAE2013-11070.
- [31] P. Birajdar, R. Taylor, K. Habib, and R. Hossain, “Analysis of Medium-Scale Laboratory Tests on Ice Crushing Dynamics,” Offshore Technology Conference, 2016. DOI: 10.4043/27482-MS.

- [32] P. Birajdar, R. Taylor, and R. Hossain, “Analysis of the Effect of Structural Compliance During Medium-Scale Laboratory Tests on Ice Crushing Dynamics,” San Francisco, California, USA, Jun. 2017.
- [33] W. F. Smith and J. Hashemi, *Foundations of materials science and engineering*, 4th ed, ser. McGraw-Hill series in materials science. Boston: McGraw-Hill, 2005, ISBN: 978-0-07-295358-9.
- [34] R. Quey, P. Dawson, and F. Barbe, “Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 17–20, pp. 1729–1745, Apr. 2011, ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.01.002.
- [35] M. Muller and M. Gross, “Interactive Virtual Materials,” *Proc. Graphics Interface*, pp. 239–246, 2004.
- [36] K. Park and G. H. Paulino, “Computational implementation of the PPR potential-based cohesive model in ABAQUS: Educational perspective,” *Engineering Fracture Mechanics*, vol. 93, pp. 239–262, Oct. 2012, ISSN: 0013-7944. DOI: 10.1016/j.engfracmech.2012.02.007.
- [37] A. Gudmundsson, *Rock Fractures in Geological Processes*. Cambridge ; New York, 2011, ISBN: 978-0-521-86392-6.

4 | Parallel Implementation of Implicit Finite Element Model with Cohesive Zones and Collision Response using CUDA

Preface

The main goal of this chapter and the corresponding publication is to describe the implementation side of the method, i.e., the technical detail of parallel assembly of the matrix and other techniques that may be useful to researchers who implement their FEM codes. This chapter explains why an efficient implementation is desirable in the case of modeling ice samples, and how the computing performance of the simulation was measured. Several simulations were conducted on materials other than ice to highlight specific details of the simulation.

The work presented here was performed at the same time as the work on Chapter 2, and the contributions of the authors are the same. I performed conceptualization, software development, numerical tests, and draft preparation. Dr. Robert Sarracino suggested several edits throughout the text and contributed to the abstract. Because of his insightful comments and changes, the writing became more coherent and logically connected. Dr. Rocky Taylor suggested several changes to the text, restructured the manuscript, and helped to address the remarks from the reviewers. The work is published in the International Journal for Numerical

Methods in Engineering¹. Note that portions of the text were adjusted to reflect examiners' comments during thesis review process.

Abstract

The aim of this work is to efficiently implement the Park-Paulino-Roesler (PPR) cohesive zone model with the objective of creating realistic high-resolution simulations of material deformation, fracture and post-fracture behavior. Intrinsically, unstructured meshes can create more realistic fracture patterns in bulk material than structured meshes, because they give more degrees of freedom for the propagation directions of the crack. Implicit methods, stable for much larger time steps, have greater potential to model both fracture and post-fracture behavior without sacrificing speed of execution. Several technical contributions are presented, including (1) GPU-accelerated implementation of the PPR cohesive zone model, (2) efficient creation of sparse matrix structure, and (3) comparison of different unloading/reloading relations when using an implicit scheme. A potential-based collision response scheme was implemented that allows one to model the interaction of fragmented material. Several test simulations are carried out to demonstrate the flexibility of the model and its ability to reproduce different materials under various loading conditions. Benchmarking results show that most of the computational time is spent by the third-party solver library, meaning that other operations do not require optimization. The library is made available as open-source.

¹<https://doi.org/10.1002/nme.5825>

4.1 Introduction

In comparison with other methods of modeling fracture, the cohesive zone (CZ) model stands out for its ability to reproduce nonlinear fracture behavior, which allows it to model polycrystalline materials, such as metal, ceramics, rock, concrete and ice. Some formulations of the CZ model distinguish between mode I and mode II fractures as well as maintain consistent fracture energy [1]. Cohesive zones are implemented as an extension of an existing technique, most commonly of the finite element (FE) or boundary element methods. When they are introduced into the model, a number of technical challenges arise. For example, the stiffness matrix in the FE formulation may lose its property of being positive definite, narrowing the choice of algorithms able to solve the corresponding linear system. The presence of CZs increases the overall element count and affects the size of the system, making it more computationally expensive. Some CZ formulations include transcendental functions that can significantly increase the computational cost and require some sort of parallel acceleration.

One approach to dealing with the above-mentioned challenges is to keep the low node count by using low-resolution geometry. For example, CZ studies are often conducted on the representative volume element (RVE) of the material, where a small sub-volume is analyzed to infer the properties of the entire object [2]. The size of the sub-volume is such that it represents the macroscopic behavior of the heterogeneous material and fits into the limitations of the computer simulation. The object of study in RVE analysis is usually a general property of the material, such as the elastic modulus or thermal conductivity.

An alternative option is to conduct simulations on a whole object of study rather than RVE. For example, a laboratory sample may consist of 10,000 grains, each of which is modeled by a finite element technique. Attempts to run high-resolution simulations lead to technical challenges: the limitations of computing hardware, the time needed to complete the simulation, and the efficiency of the software being used. There is no single solution to these issues. For example, using a computing cluster does not increase the performance because certain computational steps perform better on a single machine.

4.1.1 Related work on parallel implementation of FEM

The work presented here was inspired by the open-source packages Vega [3], GPU-TUM [4] and SOFA [5]. In these projects GPU acceleration is implemented on top of the existing code base – an approach that has both advantages and disadvantages. On the one hand, when using such packages, users continue working within a familiar framework and benefit from parallel acceleration. On the other hand, performance is often sub-optimal due to repeated conversion between host-side and device-side data formats and due to the use of outdated instruction sets. In some cases repeated data transfers may be necessary, because certain algorithms perform better on a CPU. However, if parallel acceleration is taken into account at the beginning of code development, the resulting performance is usually better.

Some FE libraries are large and require a steep learning curve. Usually they encompass a wide range of tasks and include a large set of tools. Some packages aim to cover several

branches of physics at once or may contain multiple formulations for different materials. New libraries are being developed and published regularly, sometimes taking advantage of new hardware, applying new algorithms or addressing new problems. Various tutorials explore the formulations [6], [7] and implementations [8], [9], but no specific rules or best practices last long due to the constant development of both hardware and software. Even the criteria for evaluating the results can vary – while some algorithms are made for accuracy, others aim to improve the computation speed or encompass larger geometries. For example, in the work by Wang and Yang [10], performance improvement is achieved by introducing a parallel-friendly method, which is applicable for a particular problem – real-time modeling of elastic deformation.

Even though the finite element libraries differ, they can be catalogued according to certain features: whether they implement an implicit or explicit method, whether they assemble the linear system, and so on. In the most common implementation of a FEM, a global stiffness matrix is assembled explicitly, and the only computationally expensive tasks are (1) assembling and (2) solving the linear system. Both steps affect the performance of the FEM implementation, but the latter is usually delegated to a third-party solver. The choice of the solver also affects the implementation details: the format used for representing the linear system, and whether the data should reside on a GPU or in a host memory.

One of the benefits of using a small custom code rather than a large established library is the ability to select any available solving algorithm at the development stage. For example, GPU-

accelerated libraries (e.g. Paralution², cuSOLVER³) can resolve linear systems with a positive definite left-hand side. An advantage of using such a library is that a system can be assembled and solved entirely on a GPU without moving the data between the host and device memory. This approach works well for linear elastic solids, but for CZ formulations the matrices are usually indefinite, and parallel algorithms do not work. To obtain the best results, performance must be tested with different algorithms and matrix formats. For example, cuSOLVER (v9) works with indefinite matrices, but the computation is not GPU-accelerated and better options exist.

The assembly stage also affects the overall performance but has more room for optimization, since the developer can select the hardware, the programming language and the best approach. If the matrix is sparse, the first step is to create its structure, which is not a time-consuming procedure, but still should be implemented efficiently. Once the matrix is initialized and cleared, the assembly begins where force derivatives are additively written to its entries. When performed per-element, several elements can contribute simultaneously to forces at a shared node and create a race condition, which presents an implementation challenge [11], [12].

As illustrated in Figure 4.1, entries of the linear system are assembled additively. If a race condition is not handled properly, the competing data is overwritten and lost. Common solutions to this problem are:

- partitioning elements into separate sets, each of which does not create a race condition

²<http://www.paralution.com/>

³<http://docs.nvidia.com/cuda/cusolver/>

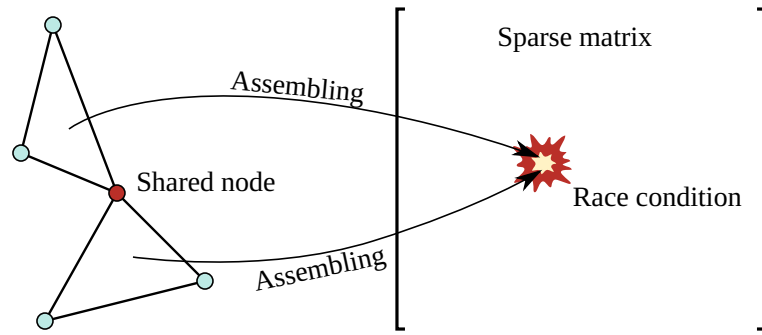


Figure 4.1: While processing the elements in parallel, a race condition occurs when two or more elements share a node.

- applying parallel computation to nodes instead of elements
- resolving the race conditions on the hardware level

The last option is the simplest and the most straightforward, as it requires little or no modifications of the sequential algorithm. Performance of this approach depends on the capabilities of the hardware. In particular, the presence of atomic addition instruction is preferable over the thread locking option.

CPU-based implementations of the Park-Paulino-Roesler (PPR) cohesive zone model [13] exist as open-source Fortran codes for ABAQUS [14], [15] and for WARP3D [16]. Both implementations are well-written, but the users should check which unloading/reloading relations are used and be aware of the possible issues [17], [18]. The flexibility of these codes is limited by the frameworks that they utilize: any issues with non-convergent solutions can be difficult to pinpoint. It is difficult to visualize the cohesive zone damage and loading curves, as the data is not easily accessible. Finally, the overall performance is limited by the implementation of the underlying library: their choice of memory management and solving algorithms.

A GPU implementation of explicit FEM with cohesive zones is described by Alhadeff et al. [19], although the code is not distributed. In their work, the GPU-efficient data format relies on the regularity of the mesh and is not intended for the general case of an irregular mesh.

4.1.2 Suggested approach

The present work aims to efficiently implement the PPR cohesive zone model with the goal of simulating high-resolution models with up to one million elements. The source code is written in C# and CUDA C. Intel MKL and the presence of a CUDA-enabled device are needed to run the library. The computationally expensive portions of the code run on the GPU; therefore the execution of the C# fragments has little effect on the performance. Benchmarks showed that about 90% of the computational time is taken up by the linear solver, the optimization of which is beyond the scope of this work. The following technical contributions are presented in this paper:

- A parallel GPU implementation is made available for processing elements and cohesive zones. The full working source code is distributed with sample setups.
- Critical implementation details are described, including the procedure for initializing the sparse matrix, the collision scheme and the memory layout in parallel portions of the code.
- The effect of different unloading/reloading relations on obtaining convergent solutions

in the implicit scheme, which is of particular interest in mixed-mode scenarios, is investigated.

To account for the interactions of fragmented material a collision detection and response scheme is implemented, which allows one to model compressive uniaxial failure, for example. Collision detection is an ongoing research topic, but few open-source solutions are readily available. This implementation uses an established approach of building and traversing a boundary volume hierarchy (BVH) during the broad phase and testing element-element intersections in the narrow phase. For response, an energy-based penalty method is applied, whose main component is finding the distance from a penetrating node to a nearest surface triangle. An existing node-triangle distance algorithm was extended to compute gradients of the distance with respect to positions of the involved nodes. Such an algorithm may be useful in a variety of other applications, and its source code is shared with this library. In addition to FEM implementation, a graphical user interface is provided that allows the user to set up geometry, adjust simulation parameters, and export rendered images.

4.2 Model formulation

This section contains the mathematical formulation and outlines the solution procedures. Nodal displacements and velocities are the variables that define the state of the system, and the goal is to determine their values at each time step. To achieve this, a system of non-linear discretized motion equations is formulated.

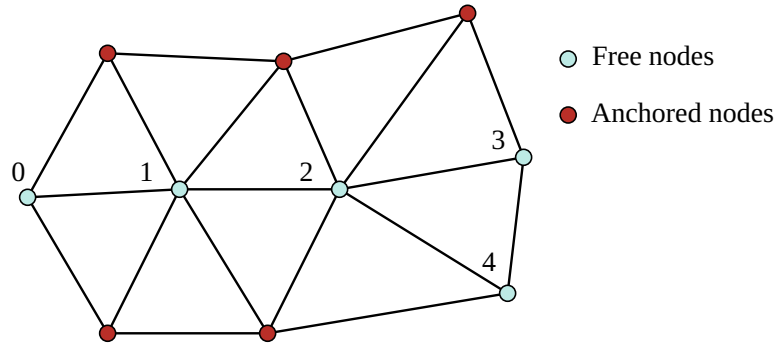


Figure 4.2: Displacements are prescribed for anchored nodes, hence their positions are known. In turn, free nodes are numbered to identify their coordinates in the resulting system of equations.

4.2.1 Discretized equation of motion and Newmark-beta method

The discretized equation of motion is formulated for the nodes, whose displacement is not prescribed explicitly (Figure 4.2). Their initial coordinates can be represented as a single vector

$$\mathbf{x}_m = [x_0, y_0, z_0, \dots, z_k]^T.$$

The displacements at time step n can be represented as

$$\mathbf{u}_n = [u_{x_0}, u_{y_0}, u_{z_0}, \dots, u_{z_k}]^T.$$

The velocities $\dot{\mathbf{u}}_n$ and accelerations $\ddot{\mathbf{u}}_n$ are defined similarly as vectors.

The motion of the non-anchored nodes follows the implicit equation

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} = \mathbf{f}_{coll} + \mathbf{f}_g - (\mathbf{f}_{elem} + \mathbf{f}_{cz}) - \mathbf{D}\dot{\mathbf{u}}_{n+1}, \quad (4.1)$$

where \mathbf{M} is the constant mass matrix and \mathbf{D} is the constant Rayleigh damping matrix. The collision and gravity forces are denoted as \mathbf{f}_{coll} and \mathbf{f}_g . The forces introduced by elements and cohesive zones are included with negative sign and denoted as \mathbf{f}_{elem} and \mathbf{f}_{cz} . The unknown variables in the equation of motion are displacements \mathbf{u}_{n+1} , velocities $\dot{\mathbf{u}}_{n+1}$, and accelerations $\ddot{\mathbf{u}}_{n+1}$. By applying the Newmark-beta integration scheme, $\dot{\mathbf{u}}_{n+1}$ and $\ddot{\mathbf{u}}_{n+1}$ are related to the unknown displacements \mathbf{u}_{n+1} ,

$$\begin{aligned} \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + (1 - \gamma)\Delta t \ddot{\mathbf{u}}_n + \gamma\Delta t \ddot{\mathbf{u}}_{n+1}, \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \frac{1 - 2\beta}{2}\Delta t^2 \ddot{\mathbf{u}}_n + \beta\Delta t^2 \ddot{\mathbf{u}}_{n+1}, \end{aligned} \quad (4.2)$$

where Δt is the time step, and β and γ are constants. The velocities $\dot{\mathbf{u}}_{n+1}$ and accelerations $\ddot{\mathbf{u}}_{n+1}$ can now be expressed in terms of the displacements \mathbf{u}_{n+1} :

$$\begin{aligned} \ddot{\mathbf{u}}_{n+1} &= \ddot{\mathbf{u}}_n \left(1 - \frac{1}{2\beta}\right) - \frac{\dot{\mathbf{u}}_n}{\Delta t\beta} + \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t^2\beta}, \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n \left(1 - \frac{\gamma}{\beta}\right) + \ddot{\mathbf{u}}_n\Delta t \left(1 - \frac{\gamma}{2\beta}\right) + \frac{\gamma}{\Delta t\beta}(\mathbf{u}_{n+1} - \mathbf{u}_n). \end{aligned} \quad (4.3)$$

Substituting expressions (4.3) into equation (4.1) yields

$$\begin{aligned} \mathbf{M} \left(\ddot{\mathbf{u}}_n \left(1 - \frac{1}{2\beta} \right) - \frac{\dot{\mathbf{u}}_n}{\Delta t \beta} + \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t^2 \beta} \right) + \\ \mathbf{D} \left(\dot{\mathbf{u}}_n \left(1 - \frac{\gamma}{\beta} \right) + \ddot{\mathbf{u}}_n \Delta t \left(1 - \frac{\gamma}{2\beta} \right) + \frac{\gamma}{\Delta t \beta} (\mathbf{u}_{n+1} - \mathbf{u}_n) \right) = \mathbf{f}, \end{aligned} \quad (4.4)$$

where

$$\mathbf{f} = \mathbf{f}_{coll} + \mathbf{f}_g - (\mathbf{f}_{elem} + \mathbf{f}_{cz}).$$

Equation (4.4) can be written simply as

$$\mathbf{F}(\mathbf{u}_{n+1}) = 0. \quad (4.5)$$

In this work the constant average acceleration method is used: $(\gamma = \frac{1}{2}, \beta = \frac{1}{4})$. It has been shown that Newmark-beta method does not conserve energy or momentum [20], [21]. Various alternatives exist, such as Hilber-Hughes-Taylor Method that has a higher numerical damping and is used in ABAQUS, for example. In this work, the load is applied to the samples up to the point of fracture. During this initial loading period, the behavior is essentially quasi-static, meaning that the whole system remains at internal equilibrium with the kinetic energy and momentum being negligible, and the use of Newmark-beta method does not result in uncontrolled growth of energy or momentum. After the fracture occurs, the elastic energy is released and the fragments begin to move in various directions. This work aims to describe

this process only qualitatively in terms of the velocities, the number of fragments, and the overall direction of motion at the early stages immediately after the fracture, when the typical problems of Newmark-beta method do not fully manifest themselves. If needed, the simulation source code can be modified to apply a different integration method.

The approximate solution to equation (4.5) is obtained iteratively via the Newton-Raphson method. The approximating sequence $\mathbf{u}^{(k)}$ is given by the recurrence relation

$$\left[\frac{\partial F_i}{\partial u_j} \right]_{\mathbf{u}=\mathbf{u}^{(k)}} \Delta \mathbf{u}^{(k)} = -\mathbf{F}(\mathbf{u}^{(k)}), \quad (4.6)$$

where $\Delta \mathbf{u}^{(k)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$. In order to apply this method, the Jacobian of \mathbf{F} must be calculated, which requires obtaining partial derivatives of forces $\mathbf{f}_{coll}, \mathbf{f}_{elem}, \mathbf{f}_{cz}$. The entries $\partial F_i / \partial u_j$ are nonzero only when nodes i and j are connected via mesh elements, cohesive elements or via collisions. For implementation this means that the matrix is sparse, and its structure may be different at each time step due to different collision geometries.

4.2.2 Elastic response

For computation of elastic forces a corotational formulation on tetrahedral elements is applied [6], which assumes small deformations, but allows arbitrary rotations. It is a linear elastic model where the rotation of each element is eliminated before the forces are computed. The initial configuration of four nodes of a tetrahedral element is described by a 12-component

vector of undeformed rest positions

$$\mathbf{x}_{el} = [x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3]^T,$$

a vector of current world-space positions

$$\tilde{\mathbf{x}}_{el} = [\tilde{x}_0, \tilde{y}_0, \tilde{z}_0, \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{x}_3, \tilde{y}_3, \tilde{z}_3]^T,$$

and the force vector

$$\mathbf{f}_{el} = [f_{x_0}, f_{y_0}, f_{z_0}, f_{x_1}, f_{y_1}, f_{z_1}, f_{x_2}, f_{y_2}, f_{z_2}, f_{x_3}, f_{y_3}, f_{z_3}]^T.$$

The force is then computed as

$$\mathbf{f}_{el} = \mathbf{R}\mathbf{K}(\mathbf{R}^{-1}\tilde{\mathbf{x}}_{el} - \mathbf{x}_{el}),$$

where \mathbf{K} is the constant stiffness matrix [22] and \mathbf{R} is the 12×12 rotation matrix of the element. Rotation can be extracted from the deformation gradient via singular value decomposition. For simplicity, only tetrahedral elements are considered in this work. If needed, the source code can be modified to use other types of elements (e.g. polyhedral), and other formulations of elastic response can be used as well. For example, large deformations can be handled by hyperelastic formulation as described by Sifakis [7].

4.2.3 Forces from cohesive elements

An extensive review of CZ models is done by Park and Paulino [1], where advantages and disadvantages of different approaches are highlighted. Park, Paulino and Roesler have also formulated their own (PPR) cohesive zone model [13], which has several desirable properties, such as distinguishing between normal and shear loads, modeling both brittle and ductile materials and having a consistent fracture energy. The normal cohesive traction $T_n(\Delta_n, \Delta_t)$ is the derivative of the potential in the normal direction, whereas the tangential cohesive traction $T_t(\Delta_n, \Delta_t)$ is the derivative in the tangential direction. The potential is defined as

$$\begin{aligned} \Psi(\Delta_n, \Delta_t) = & \min(\phi_n, \phi_t) + \\ & \left[\Gamma_n \left(1 - \frac{\Delta_n}{\delta_n} \right)^\alpha \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^m + \langle \phi_n - \phi_t \rangle \right] \times \\ & \left[\Gamma_t \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^\beta \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^n + \langle \phi_t - \phi_n \rangle \right], \end{aligned} \quad (4.7)$$

where $\langle \cdot \rangle$ is the Macaulay bracket, i.e.

$$\langle x \rangle = \begin{cases} 0 & x < 0, \\ x & x \geq 0. \end{cases}$$

Here, ϕ_n and ϕ_t are mode I and II fracture energies; α and β are non-dimensional parameters that define the shape of the traction-separation curves (should be greater or equal than one); and λ_n, λ_t are non-dimensional slope indicators that define the location of the peak normal and

tangential tractions with respect to the corresponding maximum opening distances δ_n and δ_t .

Energy constants Γ_n and Γ_t are defined as

$$\Gamma_n = (-\phi_n)^{\langle \phi_n - \phi_t \rangle / (\phi_n - \phi_t)} \left(\frac{\alpha}{m} \right)^m,$$

$$\Gamma_t = (-\phi_t)^{\langle \phi_t - \phi_n \rangle / (\phi_t - \phi_n)} \left(\frac{\beta}{n} \right)^n$$

when $\phi_n \neq \phi_t$, and

$$\Gamma_n = -\phi_n \left(\frac{\alpha}{m} \right)^m, \quad \Gamma_t = \left(\frac{\beta}{n} \right)^n$$

for the case when $\phi_n = \phi_t$. Non-dimensional exponents m and n are related to initial slope indicators λ_n and λ_t by the following expressions

$$m = \frac{\alpha(\alpha-1)\lambda_n^2}{(1-\alpha\lambda_n^2)}, \quad n = \frac{\beta(\beta-1)\lambda_t^2}{(1-\beta\lambda_t^2)}.$$

Maximum openings δ_n, δ_t can be expressed as functions of cohesive strengths [13], consequently the list of PPR parameters becomes $(\phi_n, \phi_t; m, n; \sigma_{max}, \tau_{max}; \alpha, \beta)$. The traction-separation relations are obtained by differentiating the potential function:

$$T_n(\Delta_n, \Delta_t) = \frac{\Gamma_n}{\delta_n} \left[m \left(1 - \frac{\Delta_n}{\delta_n} \right)^\alpha \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^{m-1} - \alpha \left(1 - \frac{\Delta_n}{\delta_n} \right)^{\alpha-1} \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^m \right]$$

$$\times \left[\Gamma_t \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^\beta \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^n + \langle \phi_t - \phi_n \rangle \right] \quad (4.8)$$

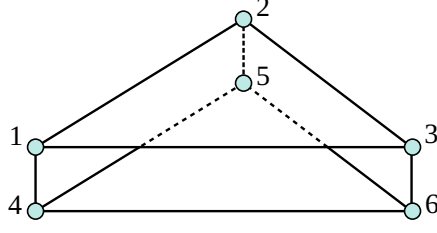


Figure 4.3: A cohesive zone that connects two triangular facets.

$$T_t(\Delta_n, \Delta_t) = \frac{\Gamma_t}{\delta_t} \left[n \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^\beta \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^{n-1} - \beta \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^{\beta-1} \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^n \right] \quad (4.9)$$

$$\times \left[\Gamma_n \left(1 - \frac{\Delta_n}{\delta_n} \right)^\alpha \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^m + \langle \phi_n - \phi_t \rangle \right] \frac{\Delta_t}{|\Delta_t|}.$$

Functions (4.8) and (4.9) are continuous and differentiable, with equal partial derivatives $\partial T_n / \Delta_t$ and $\partial T_t / \Delta_n$, which results in a symmetric matrix in equation (4.6).

Since the mesh elements are tetrahedral, cohesive zones between them are triangular and connect 3 pairs of nodes (Figure 4.3). The values of $T_n(\Delta_n, \Delta_t)$ and $T_t(\Delta_n, \Delta_t)$ vary at different points of the CZ. To obtain the average force acting across the whole cohesive zone, the traction-separation relations are integrated over the entire area. This is done numerically by taking three Gauss points that correspond to barycentric coordinates $(\frac{2}{3}, \frac{1}{6}, \frac{1}{6})$. With respect to implementation, this means that traction-separation functions and their derivatives are evaluated 3 times for each cohesive zone, which adds to computational complexity. On a CPU such computation was found to be rather slow, doubling the total runtime.

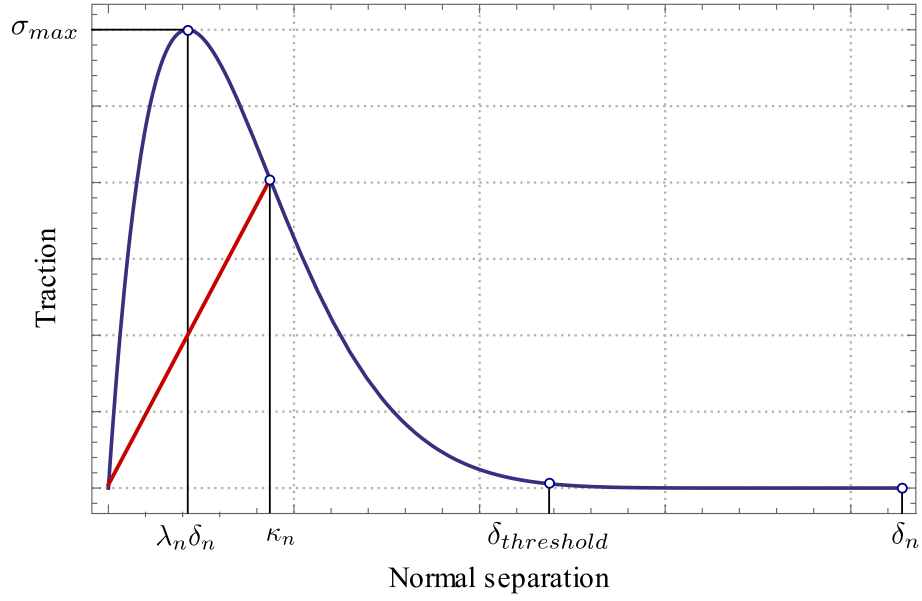


Figure 4.4: Traction-separation relation in the normal direction. The blue curve shows the loading/softening path; the red line shows the unloading/reloading path. $\delta_{threshold}$ denotes the failure criterion used at the implementation stage.

4.2.4 Unloading/reloading of CZ

To account for progressive accumulation of damage, CZ models may include history-dependent unloading/reloading relations. A linear unloading path is used frequently (Figure 4.4), although there are several ways for linear unloading/reloading relations to be formulated. In this work two options will be compared: (1) uncoupled and (2) thermodynamically consistent relations. Both formulations are based on the loading history variables

$$\kappa_n = \max\{\Delta_n\}, \quad \kappa_t = \max\{|\Delta_t|\}. \quad (4.10)$$

and are summarized in Table 4.1.

Table 4.1: Formulation of uncoupled and thermodynamically consistent unloading/reloading relations.

	Uncoupled [23]	Thermodynamically consistent [18]
Criterion	$\Delta_n < \kappa_n, \Delta_t < \kappa_t$	$\Delta_n < \kappa_n \text{ or } \Delta_t < \kappa_t$
Relations	$T_n^v(\Delta_n, \Delta_t) = T_n(\kappa_n, \Delta_t) \left(\frac{\Delta_n}{\kappa_n} \right)^{\alpha_v}$ $T_t^v(\Delta_n, \Delta_t) = T_t(\Delta_n, \kappa_t) \left(\frac{\Delta_t}{\kappa_t} \right)^{\beta_v}$	$T_n^v(\Delta_n, \Delta_t) = T_n(\kappa_n, \kappa_t) \frac{\Delta_n}{\kappa_n}$ $T_t^v(\Delta_n, \Delta_t) = T_t(\kappa_n, \kappa_t) \frac{\Delta_t}{\kappa_t}$

In the uncoupled formulation the normal and tangential relations are independent, meaning that a cohesive zone may be damaged in the normal direction, but remain intact in the tangential direction, and vice-versa. This leads to a non-symmetric matrix of force differentials $\left[\frac{\partial F_i}{\partial u_j} \right]$, which is an undesirable property, because the corresponding linear systems take more computational effort.

In the open-source Fortran implementation [14], the matrix is made symmetric by taking the average of $\partial T_n^v / \Delta_t$ and $\partial T_t^v / \Delta_n$. This creates a convergence problem in equation (4.6), leading to time step reduction and preventing the simulation from advancing in time. This effect becomes evident when some cohesive zones enter the unloading/reloading stage, for example when a sample passes the failure point on the macroscopic level.

This problem is mitigated in the thermodynamically consistent formulation, where the criterion affects both normal and tangential relations. As long as $\Delta_n \neq \kappa_n$ and $\Delta_t \neq \kappa_t$, then

$$\frac{\partial T_n^v}{\partial \Delta_t} = \frac{\partial T_t^v}{\partial \Delta_n} = 0, \quad (4.11)$$

which results in a symmetric matrix. When either $\Delta_n = \kappa_n$ or $\Delta_t = \kappa_t$, equality (4.11) does not

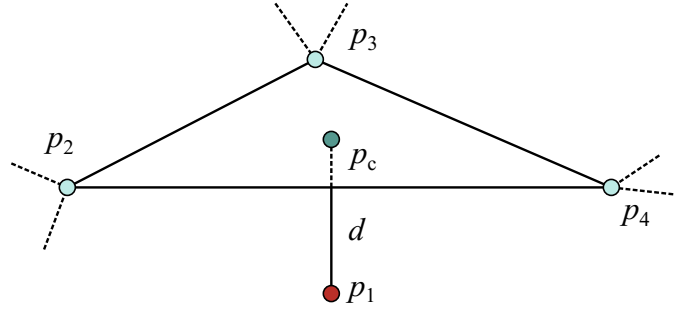


Figure 4.5: Nodes involved in collision response. A penetrating point p_1 is located below the surface of another object. The closest surface point p_c belongs to surface triangle p_2, p_3, p_4 .

hold, since κ_n, κ_t are no longer constant. At the implementation stage we can set to zero the matrix entries that correspond to $\partial T_n^v / \Delta_t, \partial T_t^v / \Delta_n$ with a hope that they are small enough to not affect the convergence of the Newton-Raphson iterations.

For illustration purposes it is helpful to define a "damaged" cohesive zone. A total count of such cohesive zones would reflect the overall damage in the sample. A CZ is called damaged if either $\kappa_n > \lambda_n \delta_n$ or $\kappa_t > \lambda_t \delta_t$. The presented formulations (Table 4.1) are independent of the damage state. However, in earlier works [14], [23] the unloading/reloading relations are applied only after the damage state is reached. In implicit methods such formulation has the worst performance, as it causes convergence problems during Newton-Raphson iterations.

4.2.5 Collision response

The collision model is formulated as a potential-based response acting on vertices that are located inside other tetrahedra (Figure 4.5). For a penetrating point p_1 the closest point p_c is found on the surface of the object and lies on a facet formed by vertices p_2, p_3, p_4 . The

interaction potential is selected as

$$\Psi = \frac{kd^2}{2}, \quad (4.12)$$

where d is the distance between points p_1 and p_c , and k is a constant. Denoting the coordinates of points p_1, p_2, p_3, p_4 by a 12-component vector

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T,$$

the corresponding force vector becomes

$$\mathbf{x} = \mathbf{f} = \nabla_{\mathbf{x}} \Psi.$$

In index notation:

$$f_i = \frac{\partial \Psi}{\partial x_i} = kd \frac{\partial d}{\partial x_i}. \quad (4.13)$$

At the implementation stage, the distance d is first obtained as its square $s = d^2$. The derivatives are also obtained of the squared distance in the form $\frac{\partial s}{\partial x_i}$. Therefore, the forces (4.13) and the force derivatives are formulated in terms of s and $\frac{\partial s}{\partial x_i}$:

$$\begin{aligned} f_i &= k\sqrt{s} \frac{\partial \sqrt{s}}{\partial x_i} = \frac{k}{2} \frac{\partial s}{\partial x_i}, \\ \frac{\partial f_i}{\partial x_j} &= \frac{k}{2} \frac{\partial^2 s}{\partial x_i \partial x_j}. \end{aligned} \quad (4.14)$$

This collision scheme does not distinguish between the contact forces from the cohesive ele-

ments and those from equations (4.14). Since both forces serve the same purpose – to prevent interpenetration of the solid material – it is safe to combine them additively. In a typical scenario when two parallel faces collide parallel to each other, both forces usually have the same direction. If such an interaction occurs at a sharp edge, then the direction of the collision forces and CZ forces will differ for the points near that edge. To minimize numerical errors it is preferable to adjust the coefficient k so that both forces have the same order of magnitude.

4.3 Implementation

The simulation loop in the present work is similar to other FE codes (Figure 4.6). The loop starts by selecting a time step and positioning the anchored nodes as prescribed by the boundary conditions. The non-linear equation $\mathbf{F}(\mathbf{u}_{n+1}) = 0$ (Eq. 4.5) is then solved to determine the motion of unconstrained nodes. Finally, the obtained solution is analyzed to prevent convergence issues. If no solution is found, the time step is reduced and the process is repeated. The active time step is expressed as:

$$\Delta t_n = \frac{\Delta t_0}{2^{\zeta_n}}, \quad (4.15)$$

where Δt_0 is the initial time step and ζ_n is the time step reduction exponent between 0 and 15.

Referring to Figure 4.6, most of the complexity is involved in solving $\mathbf{F}(\mathbf{u}_{n+1}) = 0$. This step prepares and solves equation (4.6) iteratively to find the approximation of the solution \mathbf{u}_{n+1} .

At each iteration, several computationally intensive tasks are performed:

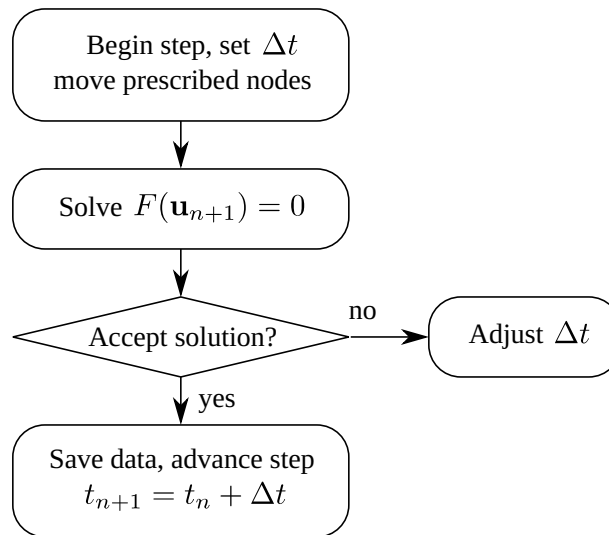


Figure 4.6: Flow diagram for a single simulation step.

- Collision detection
- Creation of sparse matrix structure
- Assembling forces from elements and cohesive zones
- Checking CZ damage accumulation
- Assembling collision forces
- Solving the linear system

The remaining part of this section is dedicated to providing the details about the implementation of the above tasks.

4.3.1 Sparse matrix format

Some FEM codes use their own formats for storing sparse matrices, which are later converted to a solver-accepted format. In the present implementation, conversion is avoided altogether, and block compressed sparse row (BCSR) format is used at all stages. BCSR is similar to the more common CSR (Figure 4.7) with the difference that each entry $v_i^{(j)}$ is itself a square matrix written sequentially into the array of values. Since our problem is three-dimensional, the submatrices have the size 3×3 . Similarly to CSR, two integer arrays define the structure of the matrix. BCSR matrices do not trigger additional conversion in the PARDISO solver and generally perform better than CSR.

The sparse structure can be viewed as a set of pairs (i, j) , where the indices i and j correspond to nodes, which interact via elastic elements, cohesive zones or collisions. When creating the structure, each interaction type is processed separately, adding ij -pairs to the collection. Some pairs may be added multiple times, but should appear only once in the resulting set. To perform this accumulation efficiently, the data structure must have $O(1)$ complexity for insertion and allow only unique elements. In C# such collection is implemented by `HashSet` class, and in C++ one would use `unordered_set`.

Subsequently, the resulting collection is converted to CSR structure arrays (Figure 4.7). A mapping of the form $(i, j) \rightarrow (\text{offset})$ is also needed at the assembly stage, where 'offset' is the location of the value entry that corresponds to the pair of interacting nodes (i, j) . Such

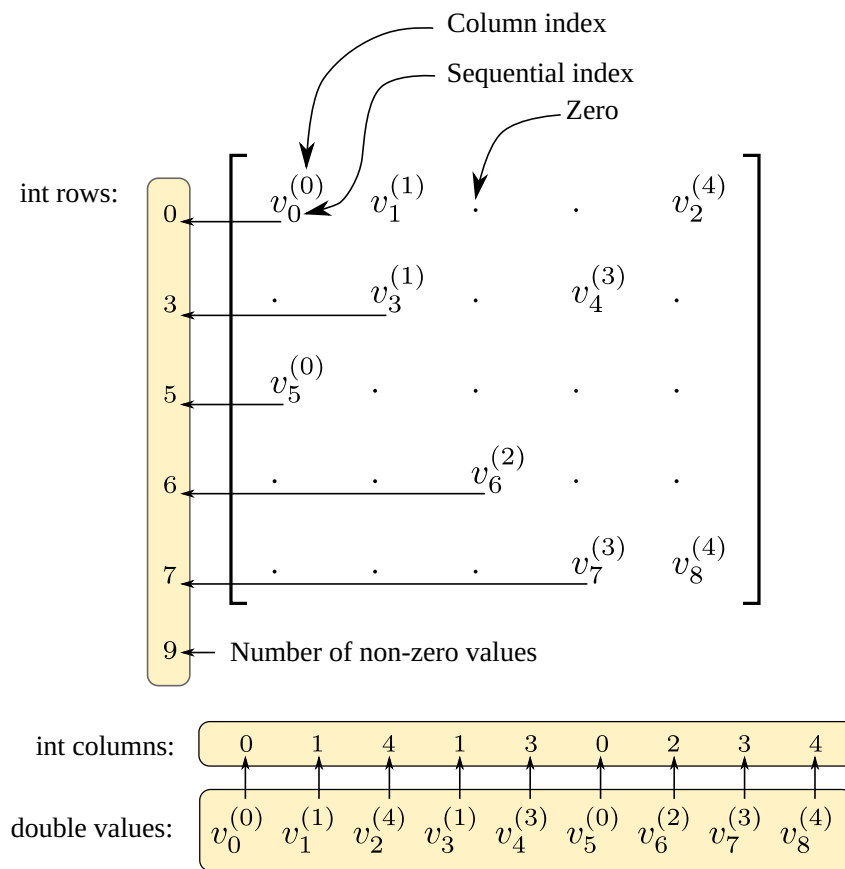


Figure 4.7: CSR matrix is defined by arrays: rows, columns and values.

a mapping is most optimally implemented by an associative array, which has $O(1)$ retrieval complexity. The overall approach is to use the most suitable data structures at each stage of the computation.

4.3.2 GPU-specific optimizations

Calculations outlined in section 4.2 are computationally intensive. Force derivatives for a single cohesive zone contain 18×18 entries, each of which is computed at 3 Gauss integration points and require 10-15 exponentiations. Benchmarking shows that it takes as much time to assemble a matrix for a model with 10,000 as it does to solve the resulting linear system. While improving the performance of the solver is beyond the scope of the present work, the assembling stage has been optimized by computing in parallel. In the present implementation, the computations are carried out using CUDA-enabled Graphics Processing Units (GPUs).

A useful hardware capability that makes the GPU instruction set different from that of popular CPUs is the presence of native atomic add operation for 64-bit floating point data. On CUDA devices with compute capability 6.x, the `atomicAdd` function is available as a single instruction call, whereas for earlier devices the same functionality is achieved through atomic compare and swap instruction `atomicCAS`. This hardware capability is a straightforward solution to resolving race conditions for parallel assembly.

It was observed that compute kernels are mostly memory-bound, and optimizing the access to global memory improves the performance. Thus, it is preferable to recompute temporary

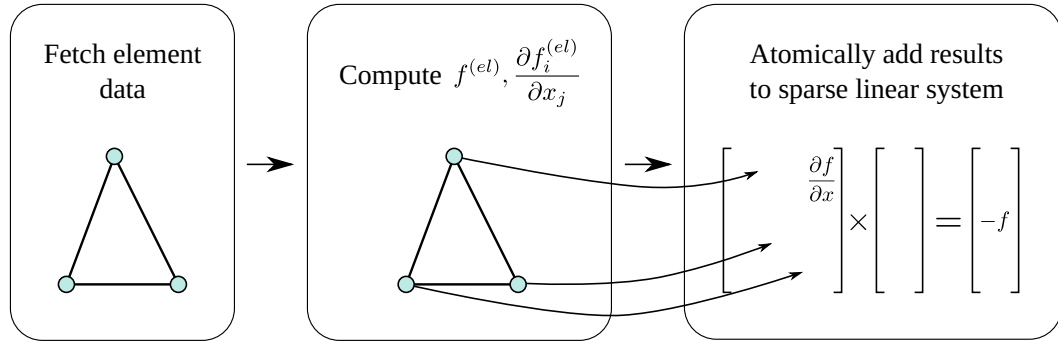


Figure 4.8: Per-element computation consists of fetching data from global memory, computing entries of the linear system and writing the results into global memory.

variables rather than fetch them from memory. Such variables include element volume, tentative velocity, acceleration, and various coefficients for elastic force computation. In addition, device-side data is stored in Structure of arrays (SoA) layout⁴ to increase the memory throughput for two-dimensional arrays. Further discussion of this topic with application to implicit FEM can be found in Allard et al. [11]. These common optimization techniques require no special effort, and kernel performance improvement is about 10%.

When adding the force derivatives $\partial f_i / \partial x_j$ to the entries $F_{i,j}$ of the stiffness matrix, the result is written at a certain memory offset into the array of values of the sparse matrix (Figure 4.8). The offsets for each (i, j) pair are determined at the time of initialization of the structure arrays. Essentially, a mapping $(i, j) \rightarrow (\text{offset})$ is created, where the sparse keys (i, j) and their corresponding 'offset' values are stored in an associative array. Unfortunately, retrieval from such an array includes branching code, which performs poorly on a GPU. To avoid divergent code, this lookup operation is performed on the CPU. Before parallel kernels are invoked, the sparse mapping of the form $(i, j) \rightarrow (\text{offset})$ is converted into the dense mapping

⁴https://en.wikipedia.org/wiki/AOS_and_SOA

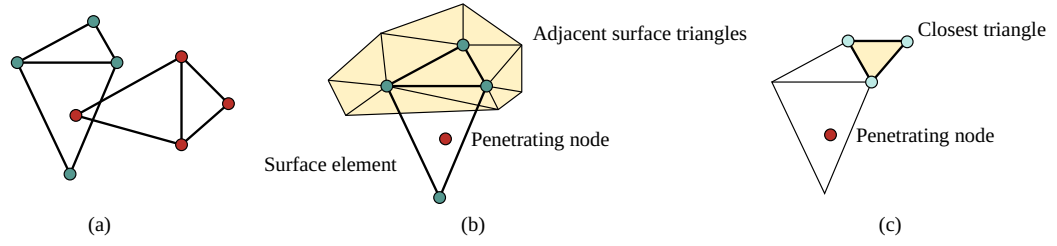


Figure 4.9: Stages of collision detection and response: (a) detecting element-element intersection, (b) searching for the closest surface triangle among the adjacent ones, (c) computing the distance to the closest triangle and the penalty forces between the interacting nodes.

$(\text{element}, \text{node}) \rightarrow (\text{offset})$, which is stored in a two-dimensional array and has non-branching retrieval. This solution is rather straightforward, but other options are possible and may be considered by researchers who implement their own code.

4.3.3 Implementation of collision detection and response

Fragmented material, adjacent grains and rigid indenters are allowed to collide, and their contact forces are explicitly computed. The reaction force pairs are applied between two colliding tetrahedra – more precisely, between a penetrating node and a surface triangle. In a broad phase, potentially colliding volumes are identified via construction of bounding volume hierarchy (BVH) of discrete oriented polytopes [24]. In the narrow phase, element-element intersections are tested, establishing the colliding vertex-face pairs (Figure 4.9). Next, for each penetrating node, the closest surface triangle is found. Finally, the penalty force is computed based on the potential function (4.12).

Collision response involves four nodes: three from the surface triangle and the one penetrat-

ing node. Consequently, there are 4 force vectors, with a total of 12 components, given by expression (4.14), and the force derivatives form a symmetric 12×12 matrix. To compute it, the first and second derivatives of the point-triangle distance d must be obtained, which is not a trivial exercise, because d is a piecewise function of 12 variables with several distinct sub-domains. The open-source algorithm by David Eberly⁵ only computes the distance itself and has 7 computational branches. In order to obtain first and second derivatives, each branch of the code was extended. Derivatives were calculated precisely, without simplifications, making the result valid for any range of distances and orientations of the nodes. The extended code is distributed freely with the rest of this library and may be used in other modeling applications. It was found that Hookean potential (4.12) results in better convergence rates than higher-order potential functions. Overall, the collision response scheme works well even for the large time steps. The narrow phase and force calculation are GPU-accelerated, whereas the broad phase is sequential and is implemented in C#.

4.3.4 Detection of failed cohesive zones

A cohesive zone fails when either its normal or tangential traction reaches zero with separation being non-zero. To detect the failed state in their numerical implementation, Spring and Paulino defined a small negative value ($\varepsilon = -10^{-7}$) to be subsequently compared to the computed values of traction $T_n(\Delta_n, \Delta_t)$ and $T_t(\Delta_n, \Delta_t)$ [14]. When the traction value fell below ε (assuming that the corresponding separation was positive), the CZ was marked as failed.

⁵<http://www.geometrictools.com/>

In this work the detection is performed differently. For brittle traction-separation relations with long near-zero "tail" (Figure 4.4) it is beneficial to set a positive failure threshold, rather than a negative one. Such an approach essentially removes the near-zero portion of the traction-separation curve. The normal threshold is set to 1% of the normal cohesive strength σ_{max} , and the tangential threshold is set to 1% of the tangential cohesive strength τ_{max} . The failure condition is only triggered when either $\Delta_n > \lambda_n \delta_n$ or $\Delta_t > \lambda_t \delta_t$.

4.4 Simulation results and benchmarking

To demonstrate the applicability of the model, several uniaxial simulations are performed and compared. Three types of deformable materials are used: "compliant", "brittle" and "ductile" (Table 4.2). The compliant material has low stiffness, high Poisson's ratio and deforms visibly. Its parameters are similar to silicone rubber. The second material, referred to as "brittle", has high stiffness, and its cohesive strength in the normal direction is not equal to the cohesive strength in tangential directions. Finally, the "ductile" material has a higher fracture energy than "brittle" and its parameters α and β are set to 1.7, resulting in a different traction-separation curve. The meshes of the simulated objects were prepared using the open-source library Neper [25]. Cohesive zones were embedded between the grain boundaries – an approach known as the intrinsic CZ model [1]. The setups are available online⁶, and can be run with the open-source distribution of this library⁷.

⁶<https://goo.gl/AvZC4u>

⁷<https://github.com/igorg520b/icFlow>

Table 4.2: Materials used in the simulation.

Parameter	Notation	Compliant	Brittle	Ductile
Young's modulus	E	2 MPa	10 GPa	10 GPa
Poisson's ratio	ν	0.45	0.3	0.3
Density	ρ	1100 kg/m ³	916.2 kg/m ³	916.2 kg/m ³
Mode I and II fracture energies	ϕ_n, ϕ_t	10 J m ⁻²	30 J m ⁻²	50 J m ⁻²
Normal cohesive strength	σ_{max}	30 kPa	1.0 MPa	1.0 MPa
Tangential cohesive strength	τ_{max}	30 kPa	1.5 MPa	1.5 MPa
Shape of traction-separation curve	α, β	3	3	1.7
Non-dimensional slope indicators in PPR model	λ_n, λ_t	0.05	0.02	0.02
Magnitude coefficient for potential-based collision response	k	200	5000	5000

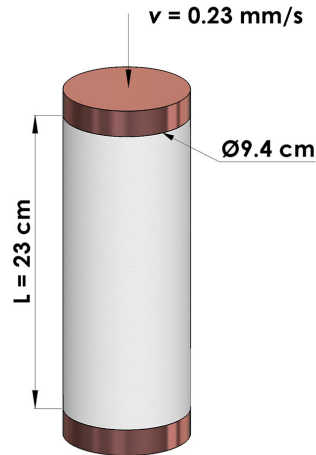


Figure 4.10: Setup of a uniaxial compression test.

Table 4.4: Mesh parameters of the cylindrical sample for compressive tests.

Height	23 cm
Diameter	9.4 cm
Grains	1,000
Elements	48,724
Cohesive zones	22,533
Nodes	26,980

For the uniaxial setups, a cylindrical sample was generated that measures 9.4 cm in diameter, 23 cm in height, and consists of 1000 grains (Table 4.4). The grains on the bottom and top surfaces are fused together to form "caps". The bottom cap is held in place, and the top cap is given a downward velocity of 0.23 mm/s (Figure 4.10).

4.4.1 Comparison of the unloading/reloading relations

In this section, two unloading/reloading relations are compared. The first test is performed with uncoupled unloading/reloading relations with damage threshold [13]. The following values are recorded: stress, time, time step reduction factor and nodal displacements. The initial time step is set to 2 s, and the shape parameters α_v, β_v (Table 4.1) are set to 1. Simulation results are shown on Figures 4.11, 4.12 and 4.13.

In the first 20 frames only elastic deformation takes place without any reduction of the time step, but starting from frame 40 the time step reduces by a factor of 2^{12} . Even though subsequent frames are computed, the simulation time does not advance significantly because the

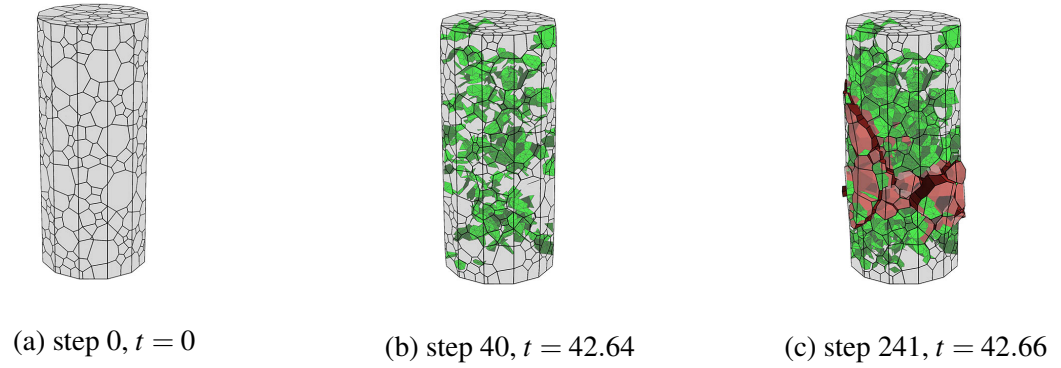


Figure 4.11: Uniaxial compression of a cylinder with uncoupled unloading/reloading. Damaged cohesive zones are shown in green and fracture surfaces are shown in brown: (a) initial configuration, (b) sample showing accumulation of damage just before failure, (c) sample at failure.

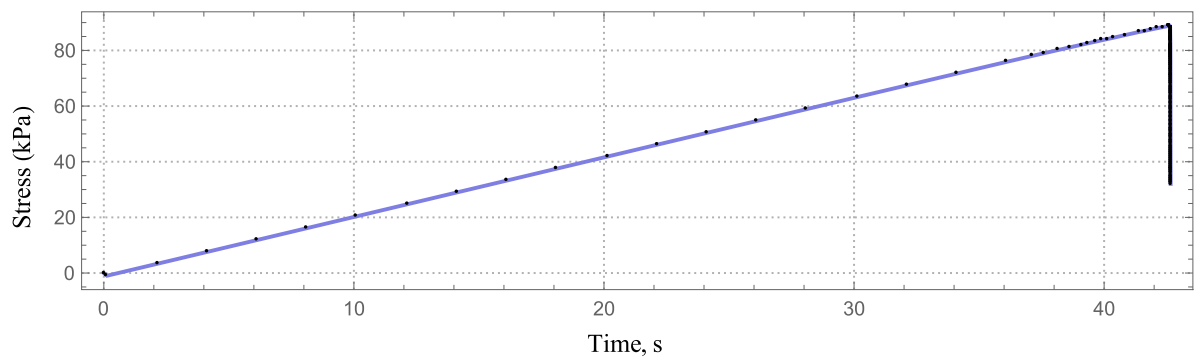


Figure 4.12: Stress vs. time. The loading curve is linear up to the point of failure.

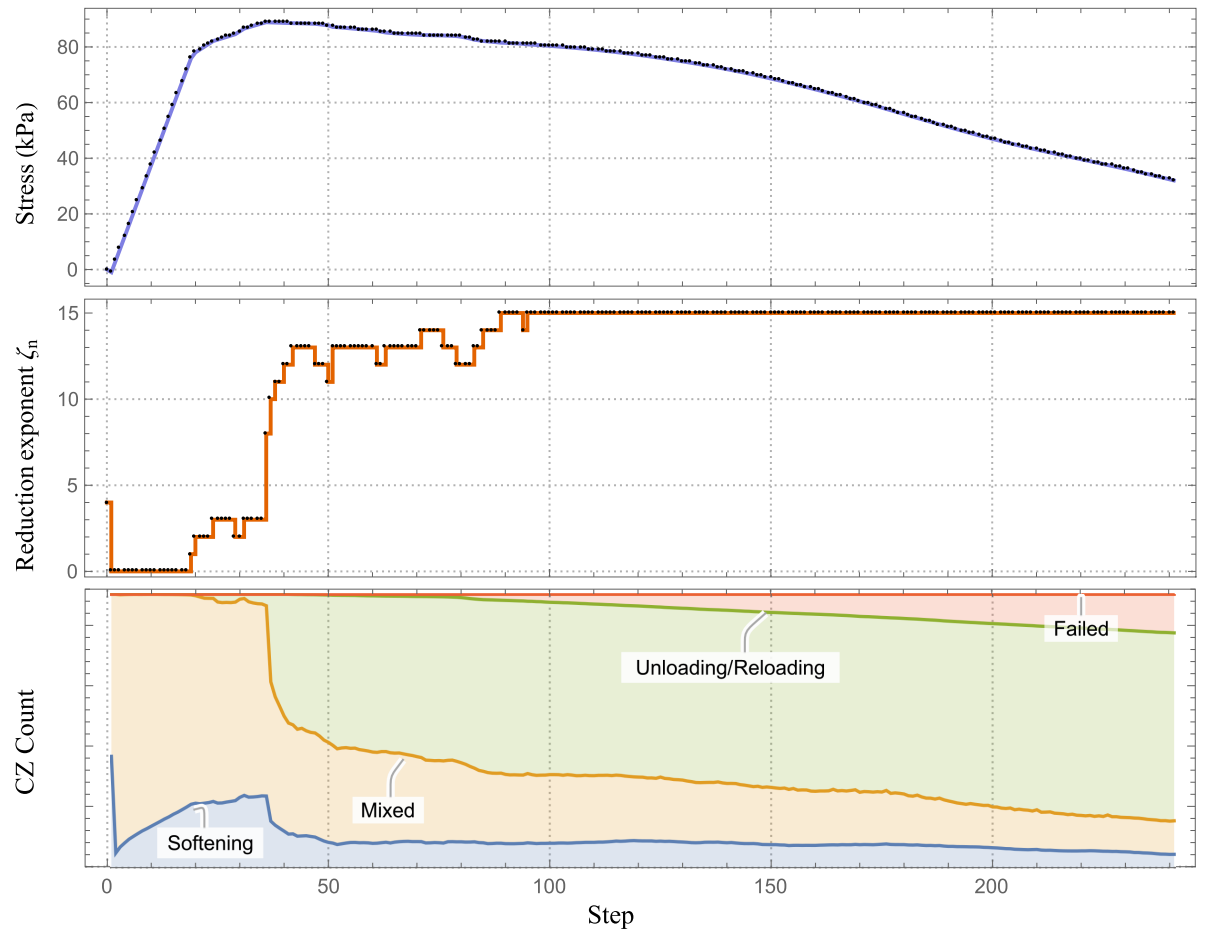


Figure 4.13: (Top) stress, (middle) reduction exponent ζ_n , (bottom) stacked plot of cohesive zones in various states vs. step number n . The reduction exponent does not return to its initial values of 0-4.

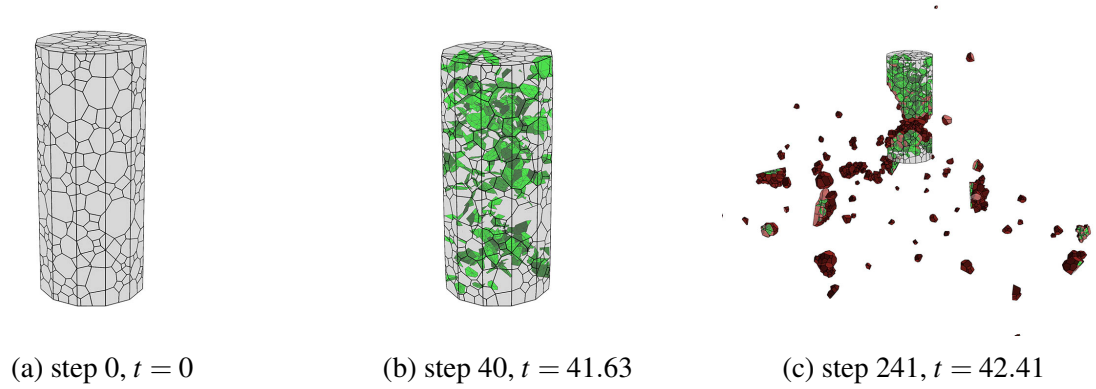


Figure 4.14: Uniaxial compression of a cylinder with thermodynamically consistent formulation: (a) initial configuration, (b) sample showing accumulation of damage just before failure, (c) ejection of fragments after failure.

steps are too small. The time span between frames 40 and 241 is only 0.02 s.

The bottom plot of Figure 4.13 shows proportions of cohesive zones in several states: softening, mixed, unloading/reloading and failed. The mixed state denotes CZs that are softening in one direction (e.g. normal) and unloading/reloading in another direction (e.g. tangential). The sudden increase of ζ_n coincides with the increase in proportion of CZs in the unloading/reloading state. It can be concluded that the formulation of the unloading/reloading relations causes poor convergence of equation (4.6). The exact problem may be the "forced" symmetrization of the gradient matrix $\left[\frac{\partial F_i}{\partial u_j} \right]$. This effect only shows up when a significant number of cohesive zones enter the unloading stage, i.e. in mixed-mode loading.

To mitigate this problem, the same test is performed with thermodynamically consistent unloading/reloading relations and no damage threshold (Figure 4.14). In this formulation, the time step gradually recovers after the non-linear event is passed (Figure 4.15), which allows the simulation to advance further beyond the point of fracture. The reduction exponent never

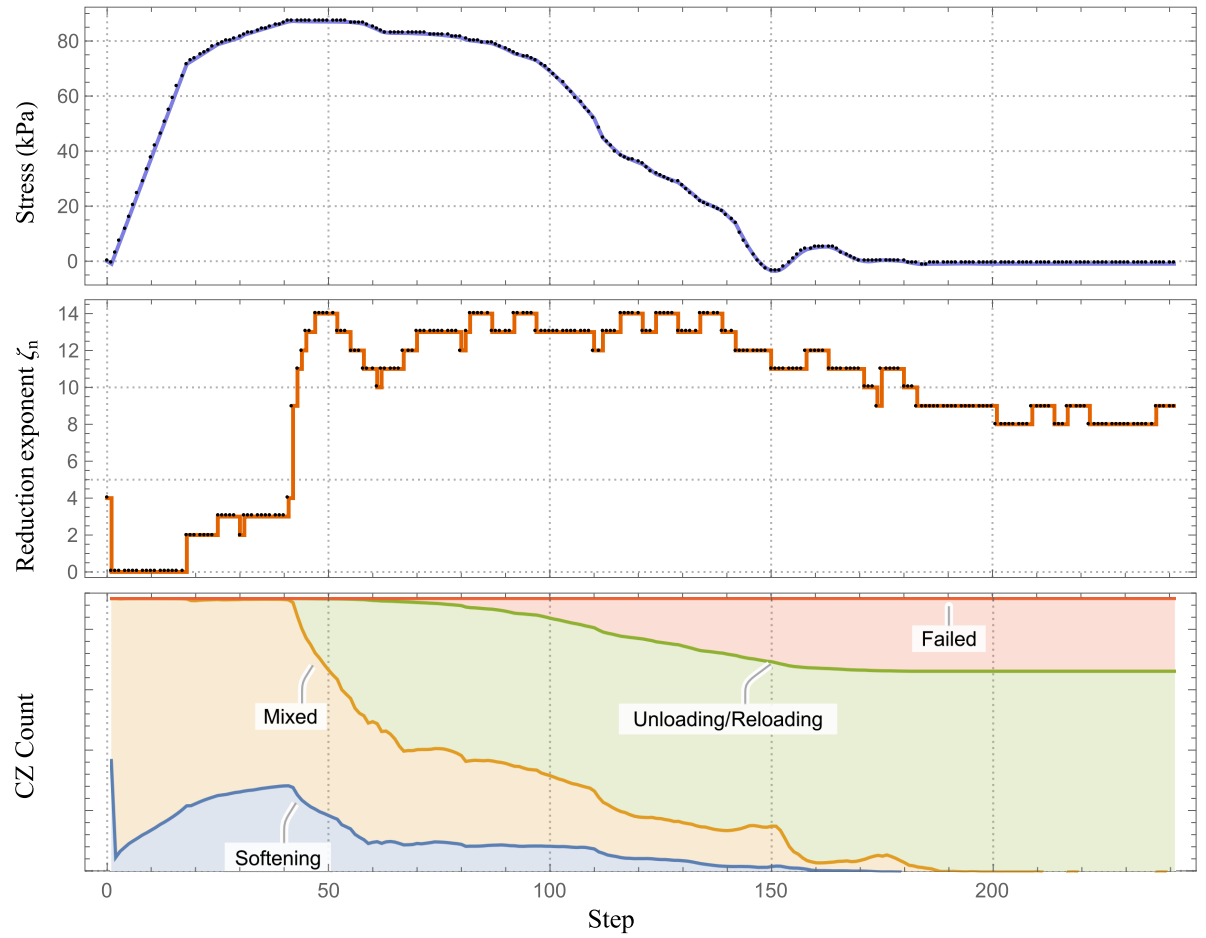


Figure 4.15: (Top) stress, (middle) reduction exponent ζ_n , (bottom) stacked plot of cohesive zones in various states vs. step number n .

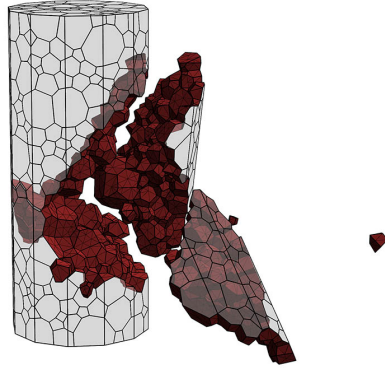
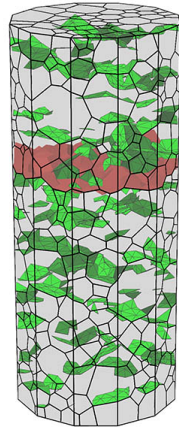


Figure 4.16: Uniaxial compression test of "brittle" material with high Young's modulus. Step 890, $t = 8.08$. Damaged CZs are not shown. Initial time step is 0.05.

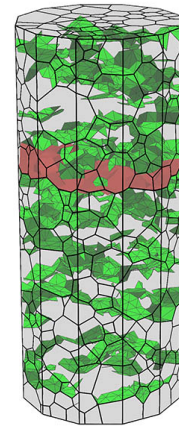
reaches its maximum value of 15, meaning that a convergent solution is obtained at each step. Some amount of non-linearity remains due to spinning of the fragments, during which the nodes follow spiral-like trajectories. If needed, such fragments can be removed from the simulation when they travel a certain distance. After the fracture completes, the remaining cohesive zones enter the unloading/reloading state (Figure 4.15). This does not affect the convergence of equation (4.6), making this formulation suitable for the implicit scheme.

One advantage of the implicit method is the ability to model deformable materials with high stiffness, such as metals and ice. This approach has been recently applied by Gribanov et al.⁸ to model the experimental results of Schulson and Lee [26] for fracture of polycrystalline ice under uniaxial loading (see also Taylor and Jordaan [27] for a discussion of modeling ice fracture during edge indentation). A uniaxial simulation can be performed with rigid material, resulting in a characteristic diagonal fracture with few fragments (Figure 4.16).

⁸Gribanov I, Taylor R, Sarracino R. Cohesive Zone Micromechanical Model for Compressive and Tensile Failure of Polycrystalline Ice. Under Review.



(a) step 85; $t = .60$



(b) step 85; $t = .37$

Figure 4.17: Uniaxial tensile test for (a) brittle; (b) ductile cohesive zones. Damaged CZs are shown in green; failed CZs are shown in brown color.

4.4.2 Comparison between brittle and ductile CZs

The difference between the brittle and the ductile behaviors is best shown in a tensile test, in which the cylindrical sample is loaded uniaxially with the negative indentation rate of -0.23 mm/s (Table 4.4). The initial time step is set to 0.05 s . In both "brittle" and "ductile" materials Young's modulus is set to 10 GPa , meaning that the elastic deformation will not be visible on the renderings. Ductile response comes only from the cohesive zones, whereas the elasticity of the elements is strictly Hookean as outlined in section 4.2.2. Both simulations begin from the initial "pristine" state and result in the horizontal fracture of the sample.

Figure 4.18 shows the adjustment of the time step throughout the simulation. The highest non-linearity occurs when the crack propagates – a process that is accurately resolved with

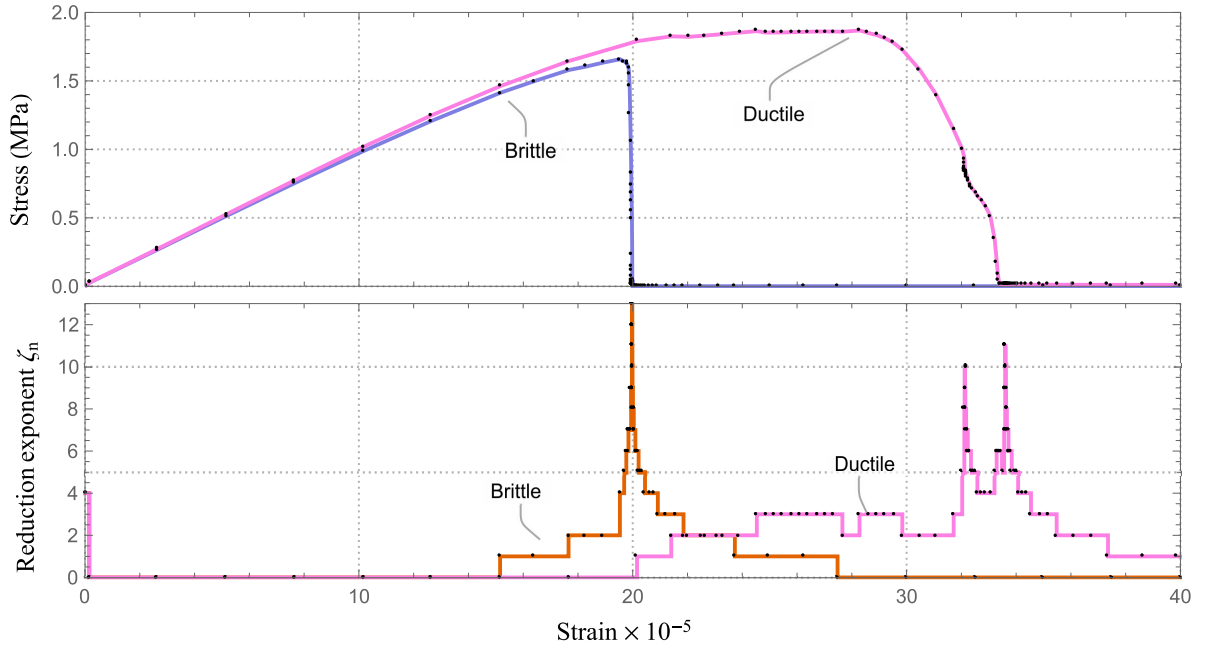


Figure 4.18: (Top) stress-strain curves and (bottom) time reduction exponent for brittle and ductile tensile tests.

reduced time steps. Once the non-linear event is passed, the time steps recover back to their initial values. The cohesive zones are less damaged in the brittle case (Figure 4.17a) than in the ductile case (Figure 4.17b). The main difference is illustrated on stress-strain curves where one of the paths has a pronounced ductile area.

Several PPR parameters control the ductile behavior. The shape of the traction-separation curve is adjusted accordingly by setting $1 < \alpha < 2$ and $1 < \beta < 2$. The closer these values are to 1, the higher the magnitudes of $\partial T_n^v / \Delta_n$ and $\partial T_t^v / \Delta_t$. At some point, high magnitudes of these derivatives affect the precision of floating-point calculations. Also, since more energy is absorbed during ductile fracture, the energy parameters ϕ_n and ϕ_t should be set higher than in brittle material.

In the brittle case where $\alpha > 5$ and $\beta > 5$, a long near-zero "tail" is observed on traction-separation curves (Figure 4.4). In practice this means that when a fracture occurs, the corresponding cohesive zones are not marked as "failed" and the newly formed surfaces are not rendered. This issue is addressed via a threshold approach as described in Section 4.3.4.

4.4.3 Performance analysis

One of the goals of the current work is to create an efficient implementation which would allow one to represent full-sized laboratory specimens. While the parallel code is implemented in CUDA C, the graphical user interface and the simulation loop are written in C#. Most computationally intensive tasks are performed either on a GPU or by the solver library. For benchmarking purposes a few additional tests were performed with more complex geometries. All simulations were performed on an i5-7600K processor running at 5.0 GHz with DDR4 RAM running at 3000MHz and an NVIDIA Quadro M5000 GPU running at stock frequency. The results are summarized in Table 4.6. To identify performance bottlenecks, the running time of individual computational steps was recorded. Table 4.7 briefly describes which steps were recorded and how they are implemented: whether they are open- or closed-source, and whether they use a CPU or a GPU.

Distribution of the computational effort between the individual steps depends mainly on the

Table 4.6: Running time of the simulations.

Simulation	Nodes	Elements	CZs	Steps	Compute time
cyl1k_comp_uncoupled	26,980	48,742	22,533	350	37 min
cyl1k_comp_therm	26,980	48,742	22,533	350	37 min
cyl1k_rigid	26,980	48,742	22,533	890	1 hr 3 min
beam1k_comp	27,283	47,119	22,516	350	34 min
beam1k_rigid	27,283	47,119	22,516	3000	2 hr 58 min
beam10k_comp	283,445	494,819	247,547	263	22 hr 8 min
cyl10k_comp	282,075	501,916	248,456	111	13 hr 32 min

Table 4.7: Computation steps.

Category	Details	Device	Implementation
Solve	PARDISO function in Intel MKL.	CPU	Closed-source
Collisions	Creation, update and traversal of BVH, element-element tests, elimination of non-unique occurrences, nearest surface triangle lookup, collision response.	mixed	C#, CUDA
Internal Forces	Forces and force derivatives from elastic elements and cohesive zones.	GPU	CUDA
Sparse structure	Creation of sparse matrix structure.	CPU	C#
Discarded	Simulation step attempts that were discarded due to non-convergence or exceeding the damage/fail threshold.	mixed	-

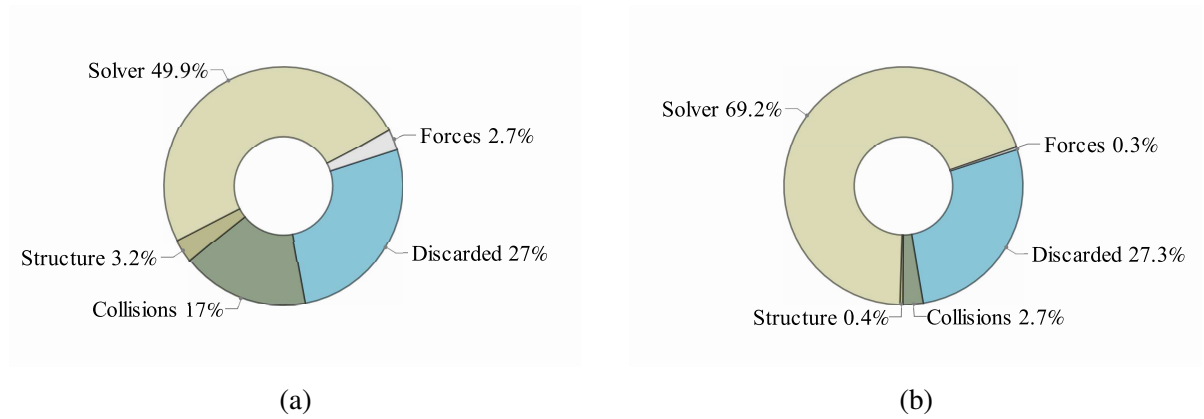


Figure 4.19: Computational effort for the simulation of (a) 1,000-grain cylinder; (b) 10,000-grain cylinder.

resolution of the meshes. Figure 4.19 shows the proportions of computational loads for the cylinders with 1,000 and 10,000 grains. In both cases the majority of the computational effort is used by the solver. The discarded computational steps are present because of the time step adjustment algorithm, which rejects the steps for which the solution cannot be obtained. For collision detection some optimization is possible, but in larger models it is less relevant. Computation of forces, which is done entirely on a GPU, takes only a small fraction of the total simulation time – between 0.3% and 2.7% – illustrating the benefit of parallel assembly.

4.5 Concluding remarks

This library was developed as a tool for modeling polycrystalline materials, such as ice under various types of loading conditions. The PPR model was chosen because of its consistent fracture energy and its ability to distinguish between normal and tangential fracture modes. After running several ABAQUS simulations, it was decided that the development of an open-source

code would allow one to better control the simulation process and model high-resolution geometries.

In many closed-source codes the time step adjustment algorithm is hidden from the user and misses the important information from the user-defined subroutines. In the presented implementation the time adjustment takes into account the damage and fail rate of cohesive zones, resulting in more accurate simulation results. Additionally, the open-source code allows one to choose any linear solver, performance of which determines the maximum resolution of the geometry and the computation speed. After thorough benchmarking, the PARDISO solver was used in this work, but other libraries can be linked with minimal changes to the code.

Because of the heavy reliance on transcendental functions, the PPR model is computationally intensive and benefits immensely from parallel programming. GPU acceleration is an important implementation feature, which speeds up the simulation by a factor of two, ultimately allowing one to perform high-resolution simulations on desktop machines.

Implementing an implicit scheme entails several technical difficulties, such as initializing the sparse matrix, obtaining the force derivatives and assembling the linear system. But the open-source implementation gives access to all simulation data and allows one to pinpoint possible problems. In this work it was noted that some unloading/reloading formulations lead to simulation halt. Based on the counts of CZ states it was shown that the problem was caused by cohesive zones in the unloading/reloading state. Several unloading/reloading formulations were compared, and the best performing formulation, which does not halt the simulation, was

selected.

Because of the ability to handle high-resolution geometry, the outlined approach can be used to simulate entire laboratory-sized specimens of metals, ceramics, rock, ice and other polycrystalline or heterogeneous materials. The presented library handles a variety of loading scenarios and a wide range of material properties. Discrete short-term events, such as fractures, are computed alongside slow loading processes. The presented investigation provides helpful information on the implementation of the PPR model and gives insight into the details of its implicit formulation.

References

- [1] K. Park and G. Paulino, “Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces,” en, *Applied Mechanics Reviews*, vol. 64, no. 6, p. 060 802, Feb. 2013, ISSN: 0003-6900. DOI: 10.1115/1.4023110.
- [2] A. Vijay, N. Paulson, and F. Sadeghi, “A 3d finite element modelling of crystalline anisotropy in rolling contact fatigue,” *International Journal of Fatigue*, vol. 106, no. Supplement C, pp. 92–102, Jan. 2018, ISSN: 0142-1123. DOI: 10.1016/j.ijfatigue.2017.09.016.
- [3] F. Sin, D. Schroeder, and J. Barbic, “Vega: Non-Linear FEM Deformable Object Simulator,” *Computer Graphics Forum*, vol. 32, no. 1, pp. 36–48, Feb. 2013, ISSN: 01677055. DOI: 10.1111/j.1467-8659.2012.03230.x.

- [4] Z. Fu, T. James Lewis, R. M. Kirby, and R. T. Whitaker, “Architecting the finite element method pipeline for the GPU,” *Journal of Computational and Applied Mathematics*, vol. 257, pp. 195–211, Feb. 2014, ISSN: 0377-0427. DOI: 10.1016/j.cam.2013.09.001.
- [5] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, “Sofa-an open source framework for medical simulation,” in *MMVR 15-Medicine Meets Virtual Reality*, vol. 125, IOP Press, 2007, pp. 13–18.
- [6] M. Muller and M. Gross, “Interactive Virtual Materials,” *Proc. Graphics Interface*, pp. 239–246, 2004.
- [7] E. Sifakis, *Finite element simulation of 3D deformable solids*. San Rafael, California: Morgan & Claypool Publishers, 2016, OCLC: 889647388, ISBN: 978-1-62705-442-3.
- [8] J. Bonet, A. J. Gil, and R. D. Wood, *Worked examples in nonlinear continuum mechanics for finite element analysis*. Cambridge: Cambridge University Press, 2012, ISBN: 978-1-107-60361-5.
- [9] G. Nikishkov, *Programming finite elements in Java™*. Dordrecht ; New York: Springer, 2010, OCLC: ocn449850661, ISBN: 978-1-84882-971-8.
- [10] H. Wang and Y. Yang, “Descent methods for elastic body simulation on the GPU,” *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–10, Nov. 2016, ISSN: 07300301. DOI: 10.1145/2980179.2980236.

- [11] J. Allard, H. Courtecuisse, and F. Faure, “Implicit FEM Solver on GPU for Interactive Deformation Simulation,” in *GPU Computing Gems Jade Edition*, Morgan Kaufmann, 2012, pp. 281–294, ISBN: 978-0-12-385963-1.
- [12] C. Cecka, A. J. Lew, and E. Darve, “Assembly of finite element methods on graphics processors,” *International journal for numerical methods in engineering*, vol. 85, no. 5, pp. 640–669, 2011.
- [13] K. Park, G. H. Paulino, and J. R. Roesler, “A unified potential-based cohesive model of mixed-mode fracture,” *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 6, pp. 891–908, Jun. 2009, ISSN: 0022-5096. DOI: 10.1016/j.jmps.2008.10.003.
- [14] D. W. Spring and G. H. Paulino, “A growing library of three-dimensional cohesive elements for use in ABAQUS,” *Engineering Fracture Mechanics*, vol. 126, pp. 190–216, Aug. 2014, ISSN: 0013-7944. DOI: 10.1016/j.engfracmech.2014.04.004.
- [15] K. Park and G. H. Paulino, “Computational implementation of the PPR potential-based cohesive model in ABAQUS: Educational perspective,” *Engineering Fracture Mechanics*, vol. 93, pp. 239–262, Oct. 2012, ISSN: 0013-7944. DOI: 10.1016/j.engfracmech.2012.02.007.
- [16] B. Healy, A. Gullerud, K. Koppenhoefer, A. Roy, S. RoyChowdhury, M. Walters, B. Bichon, K. Cochran, A. Carlyle, J. Sobo, *et al.*, “WARP3d-Release 17.2: 3-D Dynamic Nonlinear Fracture Analysis of Solids Using Parallel Computers,” *Structural Research Series (SRS)*, vol. 607, 2012.

- [17] P. Gilormini and J. Diani, “Some features of the PPR cohesive-zone model combined with a linear unloading/reloading relationship,” *Engineering Fracture Mechanics*, vol. 173, pp. 32–40, Mar. 2017, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2017.01.017.
- [18] D. W. Spring, O. Giraldo-Londoño, and G. H. Paulino, “A study on the thermodynamic consistency of the Park–Paulino–Roesler (PPR) cohesive fracture model,” *Mechanics Research Communications*, vol. 78, pp. 100–109, Dec. 2016, ISSN: 00936413. DOI: 10.1016/j.mechrescom.2016.05.006.
- [19] A. Alhadeff, W. Celes, and G. Paulino, “Mapping Cohesive Fracture and Fragmentation Simulations to Graphics Processor Units,” *International Journal for Numerical Methods in Engineering*, vol. 103, no. 12, pp. 859–893, Sep. 2015, ISSN: 00295981. DOI: 10.1002/nme.4842.
- [20] S. Krenk, “Energy conservation in Newmark based time integration algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44-47, pp. 6110–6124, Sep. 2006, ISSN: 00457825. DOI: 10.1016/j.cma.2005.12.001.
- [21] K.-J. Bathe, “Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme,” *Computers & Structures*, vol. 85, no. 7-8, pp. 437–445, Apr. 2007, ISSN: 00457949. DOI: 10.1016/j.compstruc.2006.09.004.
- [22] K.-J. Bathe, *Finite element procedures*. Englewood Cliffs, N.J: Prentice-Hall, 2014, OCLC: 930843107, ISBN: 978-0-9790049-5-7.

- [23] K. Park, “Potential-based Fracture Mechanics Using Cohesive Zone and Virtual Internal Bond Modeling,” Ph.D. thesis, University of Illinois, Urbana-Champaign, 2009.
- [24] M. Tang, D. Manocha, and R. Tong, “MCCD: Multi-core collision detection between deformable models using front-based decomposition,” *Graphical Models*, vol. 72, no. 2, pp. 7–23, Mar. 2010, ISSN: 15240703. DOI: 10.1016/j.gmod.2010.01.001.
- [25] R. Quey, P. Dawson, and F. Barbe, “Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 17–20, pp. 1729–1745, Apr. 2011, ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.01.002.
- [26] R. Lee and E. Schulson, “The Strength and Ductility of Ice Under Tension,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 110, no. 2, p. 187, 1988, ISSN: 08927219. DOI: 10.1115/1.3257049.
- [27] R. Taylor and I. Jordaan, “Probabilistic fracture mechanics analysis of spalling during edge indentation in ice,” *Engineering Fracture Mechanics*, vol. 134, pp. 242–266, Jan. 2015, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2014.10.021.

5 | The Gradient and the Hessian of the Distance Between Point and Triangle in 3D

Preface

The work presented in this chapter is essential for modeling fracture processes with multiple interacting fragments and is published in the open-access journal *Algorithms*¹, Volume 11, Issue 7. I am the primary author of the publication and performed conceptualization, software implementation, and draft writing. Co-authors Rocky Taylor and Robert Sarracino contributed by reviewing, editing and supervising. For the reader's convenience, minor formatting changes were made before the inclusion of the publication into this thesis. In particular, the constant matrices were moved from the appendix to the body of the document.

In a typical finite element method for deformable solids, the material is represented as a single mesh – a partition of a bounded connected region. Deformation of the mesh translates into nodal forces, which are in turn based on the stress-strain behavior of the material. The internal forces, however, do not prevent the material from intersecting with itself. Absence of contact response does not present a problem when modeling small deformations, but for highly de-

¹<https://www.mdpi.com/1999-4893/11/7/104>

formable solids and for simulating multiple fragments, it is desirable to introduce interaction forces for surfaces that come into contact.

A basic implementation of contact response is done by introducing a penalty force acting between the nodes of the interacting surfaces. For tetrahedral meshes, this translates to interactions between penetrating nodes and corresponding surface triangles. The magnitude of the penalty force depends on the penetration distance, and its direction depends on the relative orientation of the involved nodes. In methods with explicit integration, only the force vector itself is required, but for implicit methods, force differentials are also needed. To obtain them, one needs the gradient and the Hessian of the penetration distance.

A thorough review of relevant literature and of the existing open-source codes showed that there was no available algorithm to compute these quantities, and it was decided to develop one. Since there are many existing ways of computing the distance function, the first task was to choose the method of computing the distance itself. The main criterion is the accuracy of the computed result, as well as the ability to extend the algorithm with the computation of the gradient and the Hessian. Next, the distance formulae were differentiated twice, yielding the main results presented herein. Finally, several implementations were tested and compared for accuracy, which resulted in the C++ code that is now freely distributed. A reference implementation is included with this thesis in Appendix A.

Abstract

Computation of the distance between point and triangle in 3D is a common task in numerical analysis. The input values of the algorithm are coordinates of three points of the triangle and one point from which the distance is determined. An existing algorithm is extended to compute the gradient and the Hessian of that distance with respect to coordinates of involved points. Derivation of exact expressions for gradient and Hessian is presented, and numerical accuracy is evaluated for various cases. The algorithm has $O(1)$ time and space complexity. The included open-source code may be used in applications where derivatives of point-triangle distance are required.

5.1 Introduction

The algorithm developed by Eberly [1] evaluates the distance function between a point and a triangle in 3D. The input parameters are coordinates of four points, which are involved in the point-triangle setup. Three points are vertices of the triangle, with the remaining point being the one to which the distance is computed. The motivation for this work is the need to obtain the gradient and the Hessian of that distance with respect to the input variables. Each of the four input points possesses 3 coordinates, giving 12 independent variables for the distance function. The distance function, accordingly, has 12 first derivatives and 12×12 second derivatives, the components of the symmetric Hessian matrix.

Our approach is to differentiate the distance function provided by Eberly [1] and extend that algorithm with evaluation of first and second derivatives. The characteristics of the algorithm remain the same, i.e. the time and space complexity of the algorithm is $O(1)$. A linear array of first derivatives and a 2D array of second derivatives are added to the output. The main contribution of the authors is the derivation of the expressions for the gradient and the Hessian of the distance. A potential application of this algorithm is in finding penetration penalty forces and their differentials for colliding polygonal objects in finite element (FE) simulations.

Numerical simulations of mechanical systems often involve contact interactions, and penalty method is a common way of addressing this problem [2]. Implicit FE approaches rely on calculating derivatives of the forces generated in collisions, which in turn require first and second derivatives of the distance function. The FE simulation ARCSim [3] relies on penalty contact resolution for modeling deformation and fracture. ARCSim includes an approximation technique for obtaining the gradient and the Hessian of the distance function based on the surface normal of the interacting object. The approximation is not always accurate, which results in reduction of time steps, which sometimes halts the simulation.

In another method described by Fisher and Lin [4], the distance from the interior point to the surface of the object is precomputed on the nodes of interior polygonal mesh, representing a distance field. This discretized field allows one to estimate the spacial gradient of the distance. The accuracy is limited by the resolution of the mesh. In general, such approximation techniques are inaccurate, and there is a need for developing and utilizing the exact formula.

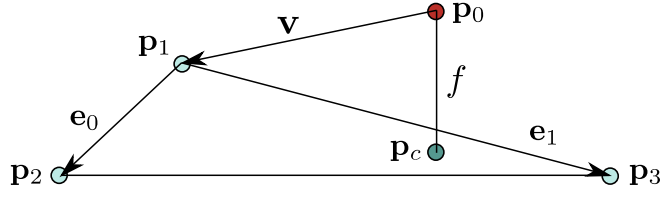


Figure 5.1: Distance f is determined between the point \mathbf{p}_0 and its projection \mathbf{p}_c onto the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. Vectors \mathbf{e}_0 , \mathbf{e}_1 and \mathbf{v} are used in calculating f .

5.2 Mathematical formulation

Figure 5.1 shows the point $\mathbf{p}_0(x_0, x_1, x_2)$ and the triangle with vertices $\mathbf{p}_1(x_3, x_4, x_5)$, $\mathbf{p}_2(x_6, x_7, x_8)$ and $\mathbf{p}_3(x_9, x_{10}, x_{11})$. The projection point \mathbf{p}_c has the barycentric coordinates $\zeta_1, \zeta_2, \zeta_3$:

$$\mathbf{p}_c = \zeta_1 \mathbf{p}_1 + \zeta_2 \mathbf{p}_2 + \zeta_3 \mathbf{p}_3 \quad (5.1)$$

The squared distance between the point and the triangle is

$$s = |\mathbf{p}_0 - \mathbf{p}_c|^2 = \sum_{k=0}^2 \left(x_k - \sum_{m=1}^3 \zeta_m x_{k+3m} \right)^2. \quad (5.2)$$

To obtain the gradient of s , expression (5.2) is differentiated:

$$\frac{\partial s}{\partial x_i} = 2 \sum_{k=0}^2 \left(x_k - \sum_{m=1}^3 \zeta_m x_{k+3m} \right) \left(\delta_{(k)(i)} - \sum_{m=1}^3 \left(\zeta_m \delta_{(k+3m)(i)} + x_{k+3m} \frac{\partial \zeta_m}{\partial x_i} \right) \right), \quad (5.3)$$

where δ_{ij} denotes the Kronecker symbol. Expressions for the barycentric coordinates ζ_m are given by Eberly [1], who introduces the following scalar coefficients:

$$a = \mathbf{e}_0 \cdot \mathbf{e}_0, b = \mathbf{e}_0 \cdot \mathbf{e}_1, c = \mathbf{e}_1 \cdot \mathbf{e}_1, d = \mathbf{e}_0 \cdot \mathbf{v}, e = \mathbf{e}_1 \cdot \mathbf{v}, \quad (5.4)$$

where $\mathbf{e}_0 = \mathbf{p}_2 - \mathbf{p}_1$, $\mathbf{e}_1 = \mathbf{p}_3 - \mathbf{p}_1$ and $\mathbf{v} = \mathbf{p}_1 - \mathbf{p}_0$. The barycentric coordinates are then

$$\zeta_1 = 1 - (\zeta_2 + \zeta_3); \zeta_2 = \frac{be - cd}{ac - b^2}; \zeta_3 = \frac{bd - ae}{ac - b^2}. \quad (5.5)$$

Expressions (5.5) are used when \mathbf{p}_c belongs to the interior region of the triangle. Cases when \mathbf{p}_c belongs to an edge or coincides with one of the vertices are discussed in the next section.

To obtain derivatives of ζ_2 and ζ_3 , expressions (5.5) are differentiated:

$$\frac{\partial \zeta_2}{\partial x_i} = \frac{1}{\Delta} \left(e \frac{\partial b}{\partial x_i} + b \frac{\partial e}{\partial x_i} - d \frac{\partial c}{\partial x_i} - c \frac{\partial d}{\partial x_i} \right) - \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_i} (be - cd), \quad (5.6)$$

$$\frac{\partial \zeta_3}{\partial x_i} = \frac{1}{\Delta} \left(d \frac{\partial b}{\partial x_i} + b \frac{\partial d}{\partial x_i} - e \frac{\partial a}{\partial x_i} - a \frac{\partial e}{\partial x_i} \right) - \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_i} (bd - ae), \quad (5.7)$$

where $\Delta = ac - b^2$ and $\frac{\partial \Delta}{\partial x_i} = a \frac{\partial c}{\partial x_i} + c \frac{\partial a}{\partial x_i} - 2b \frac{\partial b}{\partial x_i}$. Derivatives of ζ_1 are

$$\frac{\partial \zeta_1}{\partial x_i} = - \left(\frac{\partial \zeta_2}{\partial x_i} + \frac{\partial \zeta_3}{\partial x_i} \right). \quad (5.8)$$

Coefficients a, b, c, d, e (5.4) can be expanded in terms of x_i :

$$\begin{aligned}
a &= (x_6 - x_3)^2 + (x_7 - x_4)^2 + (x_8 - x_5)^2, \\
b &= (x_9 - x_3)(x_6 - x_3) + (x_{10} - x_4)(x_7 - x_4) + (x_{11} - x_5)(x_8 - x_5), \\
c &= (x_9 - x_3)^2 + (x_{10} - x_4)^2 + (x_{11} - x_5)^2, \\
d &= (x_3 - x_0)(x_6 - x_3) + (x_4 - x_1)(x_7 - x_4) + (x_5 - x_2)(x_8 - x_5), \\
e &= (x_9 - x_3)(x_3 - x_0) + (x_{10} - x_4)(x_4 - x_1) + (x_{11} - x_5)(x_5 - x_2),
\end{aligned} \tag{5.9}$$

Then, gradients of (5.9) are

$$\begin{aligned}
\left[\frac{\partial a}{\partial x_i} \right] &= 2 \times [0, 0, 0, x_3 - x_6, x_4 - x_7, x_5 - x_8, x_6 - x_3, x_7 - x_4, x_8 - x_5, 0, 0, 0], \\
\left[\frac{\partial b}{\partial x_i} \right] &= [0, 0, 0, 2x_3 - x_6 - x_9, 2x_4 - x_7 - x_{10}, 2x_5 - x_8 - x_{11}, x_9 - x_3, x_{10} - x_4, \\
&\quad x_{11} - x_5, x_6 - x_3, x_7 - x_4, x_8 - x_5], \\
\left[\frac{\partial c}{\partial x_i} \right] &= 2 \times [0, 0, 0, x_3 - x_9, x_4 - x_{10}, x_5 - x_{11}, 0, 0, 0, x_9 - x_3, x_{10} - x_4, x_{11} - x_5], \\
\left[\frac{\partial d}{\partial x_i} \right] &= [x_3 - x_6, x_4 - x_7, x_5 - x_8, x_0 - 2x_3 + x_6, x_1 - 2x_4 + x_7, x_2 - 2x_5 + x_8, \\
&\quad x_3 - x_0, x_4 - x_1, x_5 - x_2, 0, 0, 0], \\
\left[\frac{\partial e}{\partial x_i} \right] &= [x_3 - x_9, x_4 - x_{10}, x_5 - x_{11}, x_0 - 2x_3 + x_9, x_1 - 2x_4 + x_{10}, \\
&\quad x_2 - 2x_5 + x_{11}, 0, 0, 0, x_3 - x_0, x_4 - x_1, x_5 - x_2],
\end{aligned} \tag{5.10}$$

Substituting equations (5.10) into (5.6) and (5.7) allows to obtain $\frac{\partial s}{\partial x_i}$ in terms of x_i .

The second derivatives of s are procured in the same manner. First, expression (5.3) is differentiated to obtain

$$\frac{\partial^2 s}{\partial x_i \partial x_j} = 2 \sum_{k=0}^2 ((\xi')^2 + 2\xi\xi''), \quad (5.11)$$

where

$$\xi = x_k - \sum_{m=1}^3 \zeta_m x_{k+3m}, \quad (5.12)$$

$$\xi' = \delta_{(k)(i)} - \sum_{m=1}^3 \left(\zeta_m \delta_{(k+3m)(i)} + x_{k+3m} \frac{\partial \zeta_m}{\partial x_i} \right), \quad (5.13)$$

$$\xi'' = - \sum_{m=1}^3 \left(\frac{\partial \zeta_m}{\partial x_j} \delta_{(k+3m)(i)} + \frac{\partial \zeta_m}{\partial x_i} \delta_{(k+3m)(j)} + x_{k+3m} \frac{\partial^2 \zeta_m}{\partial x_i \partial x_j} \right). \quad (5.14)$$

Second derivatives of barycentric coordinates are obtained by differentiating expressions (5.6) and (5.7):

$$\begin{aligned} \frac{\partial^2 \zeta_2}{\partial x_i \partial x_j} = & \frac{1}{\Delta} \left(\frac{\partial b}{\partial x_j} \frac{\partial e}{\partial x_i} + \frac{\partial b}{\partial x_i} \frac{\partial e}{\partial x_j} - \frac{\partial c}{\partial x_j} \frac{\partial d}{\partial x_i} - \frac{\partial c}{\partial x_i} \frac{\partial d}{\partial x_j} \right) \\ & + \frac{1}{\Delta} \left(e \frac{\partial^2 b}{\partial x_j \partial x_i} + b \frac{\partial^2 e}{\partial x_j \partial x_i} - d \frac{\partial^2 c}{\partial x_j \partial x_i} - c \frac{\partial^2 d}{\partial x_j \partial x_i} \right) \\ & + \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_j} \left(d \frac{\partial c}{\partial x_i} + c \frac{\partial d}{\partial x_i} - e \frac{\partial b}{\partial x_i} - b \frac{\partial e}{\partial x_i} \right) \\ & + \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_i} \left(d \frac{\partial c}{\partial x_j} + c \frac{\partial d}{\partial x_j} - e \frac{\partial b}{\partial x_j} - b \frac{\partial e}{\partial x_j} \right) \\ & + \left(\frac{2}{\Delta^3} \frac{\partial \Delta}{\partial x_i} \frac{\partial \Delta}{\partial x_j} - \frac{1}{\Delta^2} \frac{\partial^2 \Delta}{\partial x_i \partial x_j} \right) (be - cd), \end{aligned} \quad (5.15)$$

$$\begin{aligned}
\frac{\partial^2 \zeta_3}{\partial x_i \partial x_j} = & \frac{1}{\Delta} \left(\frac{\partial b}{\partial x_j} \frac{\partial d}{\partial x_i} + \frac{\partial b}{\partial x_i} \frac{\partial d}{\partial x_j} - \frac{\partial a}{\partial x_j} \frac{\partial e}{\partial x_i} - \frac{\partial a}{\partial x_i} \frac{\partial e}{\partial x_j} \right) \\
& + \frac{1}{\Delta} \left(d \frac{\partial^2 b}{\partial x_j \partial x_i} + b \frac{\partial^2 d}{\partial x_j \partial x_i} - e \frac{\partial^2 a}{\partial x_j \partial x_i} - a \frac{\partial^2 e}{\partial x_j \partial x_i} \right) \\
& + \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_j} \left(e \frac{\partial a}{\partial x_i} + a \frac{\partial e}{\partial x_i} - d \frac{\partial b}{\partial x_i} - b \frac{\partial d}{\partial x_i} \right) \\
& + \frac{1}{\Delta^2} \frac{\partial \Delta}{\partial x_i} \left(e \frac{\partial a}{\partial x_j} + a \frac{\partial e}{\partial x_j} - d \frac{\partial b}{\partial x_j} - b \frac{\partial d}{\partial x_j} \right) \\
& + \left(\frac{2}{\Delta^3} \frac{\partial \Delta}{\partial x_i} \frac{\partial \Delta}{\partial x_j} - \frac{1}{\Delta^2} \frac{\partial^2 \Delta}{\partial x_i \partial x_j} \right) (bd - ae),
\end{aligned} \tag{5.16}$$

where $\frac{\partial^2 \Delta}{\partial x_i \partial x_j} = \frac{\partial a}{\partial x_j} \frac{\partial c}{\partial x_i} + \frac{\partial a}{\partial x_i} \frac{\partial c}{\partial x_j} + c \frac{\partial^2 a}{\partial x_i \partial x_j} + a \frac{\partial^2 c}{\partial x_i \partial x_j} - 2 \frac{\partial b}{\partial x_i} \frac{\partial b}{\partial x_j} - 2b \frac{\partial^2 b}{\partial x_i \partial x_j}$. Similarly to equation (5.8):

$$\frac{\partial^2 \zeta_1}{\partial x_i \partial x_j} = - \left(\frac{\partial^2 \zeta_2}{\partial x_i \partial x_j} + \frac{\partial^2 \zeta_3}{\partial x_i \partial x_j} \right). \tag{5.17}$$

Second derivatives of the coefficients a, b, c, d, e are constant.

$$\left[\frac{\partial^2 a}{\partial x_i \partial x_j} \right] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.18)$$

$$\left[\frac{\partial^2 b}{\partial x_i \partial x_j} \right] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.19)$$

$$\left[\frac{\partial^2 c}{\partial x_i \partial x_j} \right] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \quad (5.20)$$

$$\left[\frac{\partial^2 d}{\partial x_i \partial x_j} \right] = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.21)$$

$$\left[\frac{\partial^2 e}{\partial x_i \partial x_j} \right] = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.22)$$

The first and the second derivatives of the distance $f = \sqrt{s}$ are then expressed as:

$$\frac{\partial f}{\partial x_i} = \frac{1}{2\sqrt{s}} \frac{\partial s}{\partial x_i}, \quad (5.23)$$

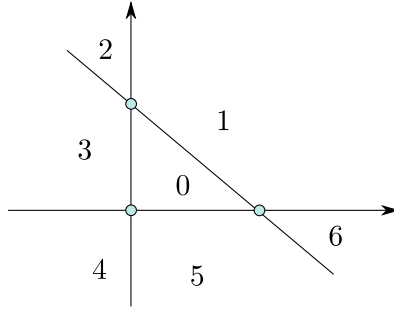


Figure 5.2: Partitioning of the plane by the triangle domain. Different domains are distinguished by the values of barycentric coordinates of the projection of \mathbf{p}_0 onto the plane of the triangle.

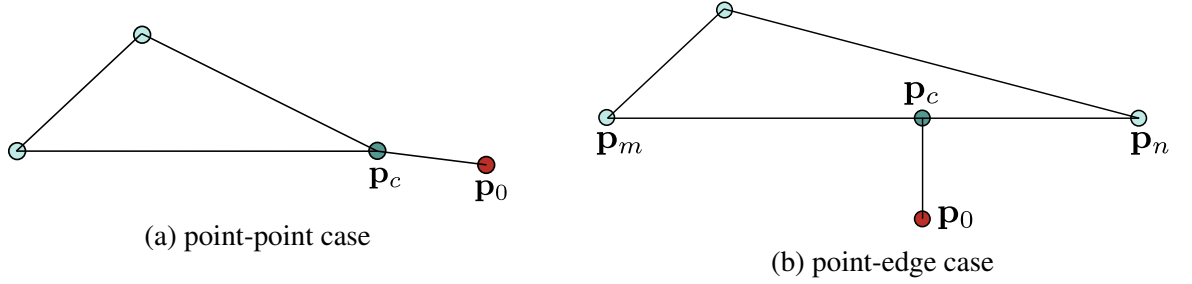


Figure 5.3: The closest point \mathbf{p}_c may coincide with (a) one of triangle's vertices or (b) belong to one of the edges.

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{1}{2\sqrt{s}} \frac{\partial^2 s}{\partial x_i \partial x_j} - \frac{1}{4s^{3/2}} \frac{\partial s}{\partial x_i} \frac{\partial s}{\partial x_j}. \quad (5.24)$$

5.3 Point-edge and point-point cases

If the projection of \mathbf{p}_0 falls outside of the interior area of the triangle (Figure 5.2), the closest point \mathbf{p}_c coincides with one of the vertices or belongs to one of the edges of the triangle. In the point-point case (Figure 5.3a), the squared distance is expressed as

$$s = |\mathbf{p}_0 - \mathbf{p}_c|^2,$$

where \mathbf{p}_c is one of $\mathbf{p}_1, \mathbf{p}_2$ or \mathbf{p}_3 . The derivatives of s with respect to x_i are obtained trivially, and the details are not discussed here.

In the point-edge case one of the barycentric coordinates of \mathbf{p}_c is zero, which simplifies expression (5.2). Let \mathbf{p}_m and \mathbf{p}_n be the vertices of the edge, to which the closest point \mathbf{p}_c belongs (Figure 5.3b). Then \mathbf{p}_c is expressed as the linear combination:

$$\mathbf{p}_c = (1 - \zeta_k)\mathbf{p}_m + \zeta_k\mathbf{p}_n,$$

where ζ_k is the non-zero barycentric coordinate of \mathbf{p}_c . This coordinate can be found as²

$$\zeta_k = \frac{(\mathbf{p}_m - \mathbf{p}_0) \cdot (\mathbf{p}_n - \mathbf{p}_m)}{|\mathbf{p}_n - \mathbf{p}_m|^2}. \quad (5.25)$$

Similarly to the previous derivations, expression (5.25) can be differentiated to obtain the first and the second derivatives of the ζ_k , which are then substituted into expressions (5.3) and (5.11).

5.4 Algorithm and testing

The calculations are implemented in double-precision floating-point arithmetic in C, and the tests are performed with squared distance s and its first and second derivatives. The first step is to determine the partition to which \mathbf{p}_c belongs (Figure 5.2). This step coincides with

²<http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html>

the original point-triangle algorithm [1], but subsequent calculations are extended to evaluate $\frac{\partial s}{\partial x_i}$ and $\frac{\partial^2 s}{\partial x_i \partial x_j}$. If \mathbf{p}_c belongs to partitions 1,3,5, then the point-line algorithm is invoked. For partitions 2,4,6, point-point calculations are performed. For partition 0 the point-plane algorithm is used.

For testing, 10 million cases were generated, about 2/3 of which are random coordinates that come from the uniform distribution in the range $[-1, 1]$. The remaining cases come from the finite element simulation of fracture, where the point \mathbf{p}_c is often close to one of the triangle's vertices. Most of the cases that come from the simulation have a low ratio of the distance to the shortest edge of the triangle, in the range between 10^{-8} and 10^{-5} . Such test cases result in a lower accuracy of the final answer than the random arrangements of points.

To evaluate the accuracy of the proposed algorithm, calculations are first performed using arbitrary precision arithmetic with at least 17 digits calculated precisely. These results are denoted $s_p, \frac{\partial s_p}{\partial x_i}, \frac{\partial^2 s_p}{\partial x_i \partial x_j}$ and are compared to the results obtained in floating-point arithmetic $s_c, \frac{\partial s_c}{\partial x_i}, \frac{\partial^2 s_c}{\partial x_i \partial x_j}$. Relative errors are computed separately for the squared distance, its first derivatives and its second derivatives:

$$E_0 = |(s_c - s_p)/s_p|, \quad (5.26)$$

$$E_1 = \max_i \left| \left(\frac{\partial s_c}{\partial x_i} - \frac{\partial s_p}{\partial x_i} \right) / \frac{\partial s_p}{\partial x_i} \right| \quad (5.27)$$

Table 5.1: Relative errors for squared distance and its first and second derivatives. The maximum values from 10 million test cases are shown.

Error measure	Value
$E_{0_{max}}$	3.78×10^{-5}
$E_{1_{max}}$	2.75×10^{-5}
$E_{2_{max}}$	1.27×10^{-3}

$$E_2 = \max_{i,j} \left| \left(\frac{\partial^2 s_c}{\partial x_i \partial x_j} - \frac{\partial^2 s_p}{\partial x_i \partial x_j} \right) / \frac{\partial^2 s_p}{\partial x_i \partial x_j} \right| \quad (5.28)$$

E_0 is the relative error of the squared distance and is used as a baseline to compare with the relative errors of the first and second derivatives E_1 and E_2 . Ideally, the values of E_0 , E_1 and E_2 would have the same order of magnitude. However, due to the large number of algebraic operations, the accuracy of the calculation of the second derivative is lower. The values E_0 , E_1 and E_2 cover a wide range of values. In about half of the cases, the precision for calculating second derivatives is better than 10^{-13} . However, the practical interest lies in investigating the worst cases, because one incorrect calculation could affect a whole scientific study.

5.4.1 Discussion

The highest errors among all test cases are shown in Table 5.1. The values come from separate test cases: $E_{0_{max}}$ is the maximum error for s , $E_{1_{max}}$ is the maximum error for $\frac{\partial s_c}{\partial x_i}$ and $E_{2_{max}}$ is the maximum error for $\frac{\partial^2 s_c}{\partial x_i \partial x_j}$. $E_{2_{max}}$ is higher than $E_{0_{max}}$ by two orders of magnitude, which is a good result, considering that it is the least accurate of 10 million tests. The tests show that

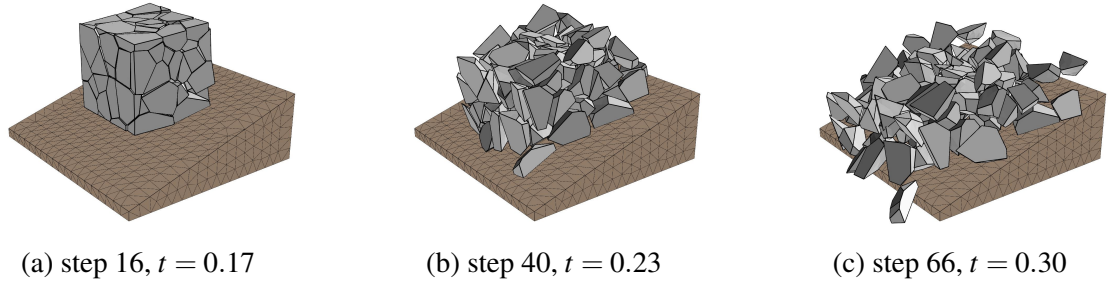


Figure 5.4: Simulation of falling and colliding grains performed with implicit FEM.

the precision is adequate for all cases, including the ones from the collision simulation.

Cases with the lowest accuracy usually correspond to the variables whose absolute values are very small, and computer simulations are often robust against such cases. For example, results of the grain interaction simulation where the current algorithm is applied are shown on Figure 5.4. The simulation advances with large time steps even when multiple fragments interact with each other.

The ratio between the distance and the largest edge of the triangle is one of the factors that affect the precision. When this ratio drops below 10^{-10} , the accuracy of the result is likely to deteriorate. The problem of the round-off error is common in numerical analysis, and should be addressed properly. If the influence of the round-off error is suspected when applying this algorithm, additional testing should be performed. In some cases, calculations can be performed with arbitrary-precision arithmetic to yield accurate results.

5.5 Conclusion

The presented algorithm has $O(1)$ time and space complexity. Only static memory allocation takes place. The algorithm has several branches that evaluate algebraic expressions sequentially, with each branch completing in constant time. The main contribution of the authors is the derivation of the exact formulae for the gradient and the Hessian of the distance function. Additionally, testing of the reference implementation was performed to ensure that adequate precision is met. The proposed algorithm may be used in applications where point-triangle distance derivatives are required. The potential future work may include precision testing for more complex cases. The C code is available as open-source³ and may be modified by the research community as needed.

References

- [1] D. Eberly. (). Distance between point and triangle in 3d, [Online]. Available: <https://www.geometrictools.com/Documentation/DistancePoint3Triangle3.pdf>.
- [2] T. A. Laursen, *Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis*. Springer Science & Business Media, 2013.

³<https://github.com/Spear520/dist>

- [3] T. Pfaff, R. Narain, J. M. De Jona, and J. F. O'Brien, "Adaptive tearing and cracking of thin sheets," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 110, 2014.
- [4] S. Fisher and M. C. Lin, "Fast penetration depth estimation for elastic bodies using deformed distance fields," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, IEEE, vol. 1, 2001, pp. 330–336.

6 | Application of Cohesive Zone Model to the Fracture Process of Freshwater Polycrystalline Ice under Flexural Loading

Preface

Beam bending tests are common in both the experimental and the numerical study of ice mechanics. This chapter describes the application of the proposed model to the four-point beam bending tests. This numerical study was performed after Chapters 2 and 3 were submitted for publication. The authors and their contributions were the same as in those chapters. I performed numerical tests and wrote the draft version of the manuscript. Dr. Rocky Taylor suggested the sources of the existing experimental data for comparison and wrote parts of the introduction and the conclusion. Dr. Robert Sarracino contributed to the analysis of the results and proofread the final version of the manuscript.

The results were presented at the conference Polar Mechanics 2018 and are published in IOP Conference Series: Earth and Environmental Science¹.

¹<https://iopscience.iop.org/article/10.1088/1755-1315/193/1/012013/meta>

From the perspective of numerical modeling, the main difference between the uniaxial tests on cylinders and the four-point bending tests on beams is in the method of applying the loads. With cylinders, displacement is applied directly to the surface of the object, as if small caps were frozen onto the ends of the cylinder. In beam bending the load is applied via contact force between the beam and four indenters. As a result, the loading curves are different from those of the uniaxial tests.

Abstract

Cohesive zone modeling is a promising technique for modeling fracture processes in polycrystalline solids. The Park-Paulino-Roesler (PPR) formulation for cohesive zones is a flexible method that allows one to reproduce behaviours of a wide range of materials. In this work, an implicit finite element simulation is performed to capture the fracture of freshwater, polycrystalline ice beams in 4-point bending tests. The distribution of the damaged cohesive zones and the indentation force are recorded as a function of time. Several simulations are performed to establish the relationship between the grain size and the flexural strength of the beams.

6.1 Introduction

While ice is a beautiful natural material, it can pose a risk to ships and structures in the marine environment, as well as to people who may use winter ice cover as a transportation route.

During the formation process, prior to establishing a level ice cover, sea ice may partially form, break up and refreeze multiple times. Its microstructural composition depends largely on its age and growth conditions. In general, sea ice contains inclusions of brine and air, often having a granular microstructure in the uppermost layer, which transitions to a columnar microstructure within a few centimeters below the top surface [1]. In dynamic ocean environments, deformation processes in the sea ice can result in more complex microstructures as the sea ice is repeatedly broken, reoriented and refrozen. Icebergs, on the other hand, calve off glacial ice masses formed from compacted snow and generally are nearly free of solid particles and soluble substances [2]. Iceberg ice typically has a polycrystalline structure that can be characterized by grain size. Over large volumes, significant variation in grain size may be expected, as well as the occurrence of pre-existing cracks which may remain in the ice or may heal over time, resulting in large veins of ice with a different microstructure in the healed crack regions [3].

In the vicinity of a large iceberg (Fig. 6.1), one can occasionally hear cracking sounds – a sign of gradual fracture processes that can culminate in sudden catastrophic splitting. For interactions between ice and structures, fracture is highly important both in limiting loads and in localizing contact into zones of intense local pressures through which the majority of loads are transmitted to a structure [4]. Consequently, understanding and modelling fracture mechanisms in such ice features is highly important. While full-scale ice structure interactions are highly complex, insights into certain aspects of ice fracture behaviour can be studied through the measurement of ice strength under controlled conditions. Such tests are performed and

experimental data is made available for compressive indentation conditions [5]–[11], as well as uniaxial compression tests on cylinders [12], [13] and flexural tests on beams [3], [14]. In the present paper, emphasis is placed on modelling freshwater polycrystalline (e.g. glacial) ice subjected to flexural loading conditions.

6.1.1 Related work on modeling

Finite element approaches to modeling damage can be roughly classified into two groups. In the first group, the damage is treated as continuous and expressed as a function of material coordinates. The crack surfaces are not explicitly tracked, and the material is not fragmented [15]. In the second group, the grains, cracks and other individual constituents are explicitly resolved. Concepts that are specific to the material, such as wing cracks, can be introduced into latter models [16]. Such approaches usually require more computational resources, because the shapes of the individual constituents are recorded and processed.

Cohesive zone models (CZM), where individual grains in the material are held together by cohesive tractions (Fig. 6.2), fall into the second category of methods. Several applications of this method to the mechanics of ice have been proposed [17]–[19].



Figure 6.1: An iceberg sighting on May 28, 2017, near Topsail Beach, Conception Bay South, NL. Photo courtesy of <http://mynewfoundlandkayakexperience.blogspot.ca>.

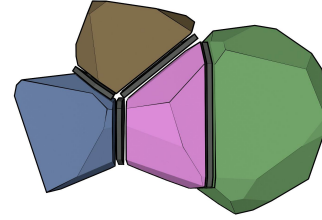


Figure 6.2: Polyhedral grains connected with cohesive zones (CZ). The rest volume of CZs can be zero or non-zero. Grains may be deformable or rigid.

6.2 Description of the method

To capture the dynamic behaviour of the modeled sample we utilize the implicit finite element method (FEM) with Newmark-beta integration. The discretized non-linear equation of motion is solved via Newton-Raphson iterations. The modeled geometry consists of separate grains which are connected by cohesive zones at the beginning of the simulation. The grains have linear elastic response, with the rotational component eliminated from the deformation gradient tensor to allow arbitrary rotations of the elements [20].

The traction-separation relations for the cohesive zones are as formulated by Park-Paulino-Roesler (PPR) [21]. All cohesive zones are triangular, with zero rest volumes. The normal separation is distinguished from the tangential separation; both normal and tangential forces

are found as the gradient of the common potential Ψ :

$$\begin{aligned} \Psi(\Delta_n, \Delta_t) = \min(\phi_n, \phi_t) + \\ \left[\Gamma_n \left(1 - \frac{\Delta_n}{\delta_n} \right)^\alpha \left(\frac{m}{\alpha} + \frac{\Delta_n}{\delta_n} \right)^m + \langle \phi_n - \phi_t \rangle \right] \times \\ \left[\Gamma_t \left(1 - \frac{|\Delta_t|}{\delta_t} \right)^\beta \left(\frac{n}{\beta} + \frac{|\Delta_t|}{\delta_t} \right)^n + \langle \phi_t - \phi_n \rangle \right], \end{aligned} \quad (6.1)$$

where $\langle \cdot \rangle$ is the Macaulay bracket, i.e.

$$\langle x \rangle = \begin{cases} 0 & x < 0, \\ x & x \geq 0. \end{cases}$$

Separation parameters in the normal direction Δ_n and tangential direction Δ_t are independent, allowing material properties in these directions to be set separately. The parameters $(\phi_n, \phi_t; \delta_n, \delta_t; \alpha, \beta; \lambda_n, \lambda_t)$ determine the properties of the material, such as fracture energy, cohesive strength and brittleness. Energy constants Γ_n and Γ_t are defined by the expressions

$$\Gamma_n = (-\phi_n)^{\langle \phi_n - \phi_t \rangle / (\phi_n - \phi_t)} \left(\frac{\alpha}{m} \right)^m, \quad \Gamma_t = (-\phi_t)^{\langle \phi_t - \phi_n \rangle / (\phi_t - \phi_n)} \left(\frac{\beta}{n} \right)^n$$

when the normal and tangential fracture energies are not equal ($\phi_n \neq \phi_t$), and

$$\Gamma_n = -\phi_n \left(\frac{\alpha}{m} \right)^m, \quad \Gamma_t = \left(\frac{\beta}{n} \right)^n$$

when they are equal ($\phi_n = \phi_t$). Non-dimensional exponents m and n are related to initial slope

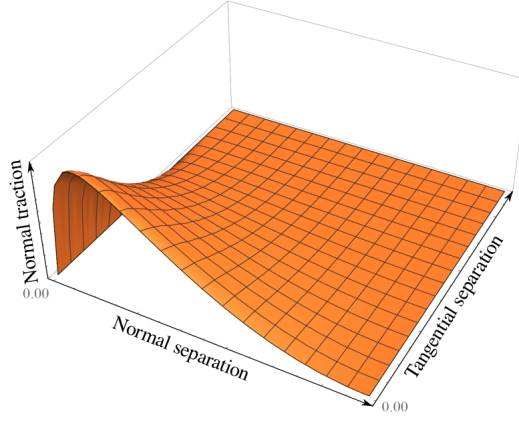


Figure 6.3: Normal component of the traction-separation relations as a function of normal and tangential separations Δ_n, Δ_t . Tangential component may be similar or different, depending on the selected parameters.

indicators λ_n and λ_t by the following expressions

$$m = \frac{\alpha(\alpha - 1)\lambda_n^2}{(1 - \alpha\lambda_n^2)}, \quad n = \frac{\beta(\beta - 1)\lambda_t^2}{(1 - \beta\lambda_t^2)}.$$

A typical traction-separation law is illustrated on Fig. 6.3.

Because of the exponential terms, the gradient of $\Psi(\Delta_n, \Delta_t)$ takes a substantial amount of time to compute, especially for a large number of CZs. For best performance, these functions are evaluated in parallel on a GPU. The matrix in the resulting linear system is indefinite symmetric and cannot be resolved by GPU-optimized algorithms, therefore a CPU-based linear solver is used. Additionally, unloading/reloading relations are implemented as described by Spring [22], as in the implicit method they perform better than those described by Park et al. [21].

To account for the interaction of fragmented material, the contact detection and response is implemented, which is based on the penetration distance. The implementation is distributed

as open-source².

6.3 Results and discussion

The flexibility of the PPR traction-separation laws allows to tune the model to a wide range of materials, including ice. A set of parameters (Table 6.1) was selected to match the experimental data for uniaxial tensile and compressive tests [19]. Some of the parameters have physical meaning, i.e. fracture energy and cohesive strength. Other parameters, such as α, β, λ_n and λ_t , determine whether the fracture will be brittle or ductile.

Table 6.1: Parameters of the material.

Parameter	Notation	Value
Young's modulus	E	10 GPa
Poisson's ratio	ν	0.3
Density	ρ	916.2 kg/m ³
Mode I and II fracture energies	ϕ_n, ϕ_t	30 J m ⁻²
Normal cohesive strength	σ_{max}	1.0 MPa
Tangential cohesive strength	τ_{max}	1.5 MPa
Shape of traction-separation curve	α, β	3
Non-dimensional slope indicators in PPR model	λ_n, λ_t	0.02
Magnitude coefficient for potential-based collision response	k	5000

The four-point bending simulations are performed on 28 different beam models. The geometries with random arrangements of grains were generated in Neper [23], and the dimensions

²<https://github.com/igorg520b/icFlow>

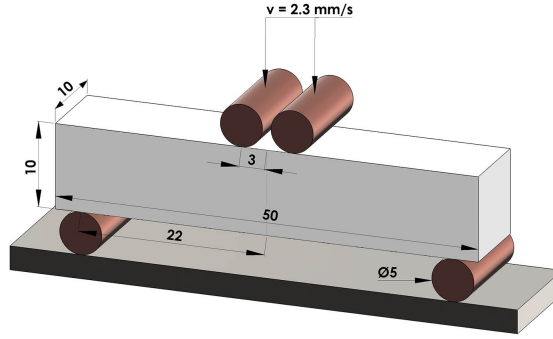


Figure 6.4: Setup of the four-point beam bending. Dimensions are shown in centimeters.

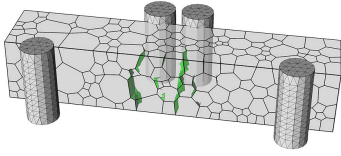


Figure 6.5: Simulation step 193; $t = 4.2797$ s. Damaged CZs shown in green, fracture surface shown in red.

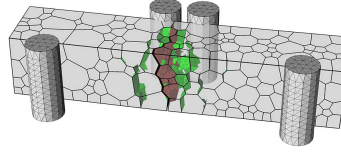


Figure 6.6: Simulation step 221; $t = 4.5106$ s.

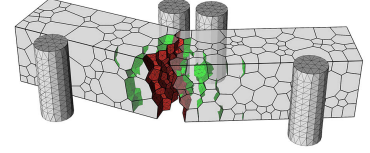


Figure 6.7: Simulation step 256; $t = 4.5216$ s.

are summarized on Fig. 6.4. The beams measure $10 \times 10 \times 50$ cm and consist of 500, 750, 1000, 1250, 1500 and 2000 grains. Each grain is subdivided into tetrahedral elements, having 100 elements per grain on average. The indenters measure 5 cm in diameter, are rigid, and move with the speed of 0.23 cm/s. The initial time step is 0.025 s. All initial setups are available along with the open-source implementation of this project.

Nodal displacements and the forces acting on the indenters are recorded at each step. Simulations perform 300 steps, passing the point of beam failure. One of the tests is illustrated in Figs. 6.5-6.7, where the initiation and propagation of the crack is shown in a beam with 500 grains. The corresponding force-time curve is nonlinear (Fig. 6.8), with the slope gradually

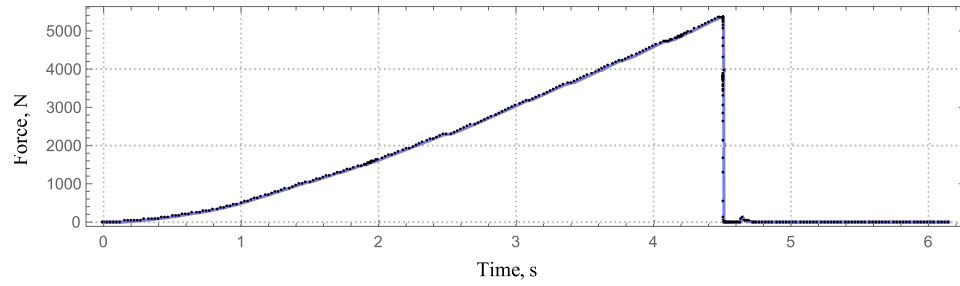


Figure 6.8: Indentation force vs time.

increasing. Displacements that occur during the loading stage are small and not visible on the images. However, the strain energy is high and the accumulation of damage in the cohesive zones is visible. In this test, the force on the indenters reaches 5349 N at the moment of fracture.

After the fracture occurs, the newly formed fragments acquire substantial velocity due to the release of elastic strain energy. Before they start moving in opposite directions, an impact occurs along the formed boundary near the indenters. In the current model, only intergranular fracture and elastic deformation are considered. In nature, however, ice would be expected to also experience local crushing at the contact points. The interplay between fracture and crushing during full-scale compressive ice-structure interactions is highly important for modelling loads of interest for engineering design, but is beyond the scope of the present paper. Accounting for such mechanisms to better model interactions involving progressive failure, which occurs during processes such as indentation, is the direction of the future work.

In each performed test the maximum force is determined before the beam fails. The corre-

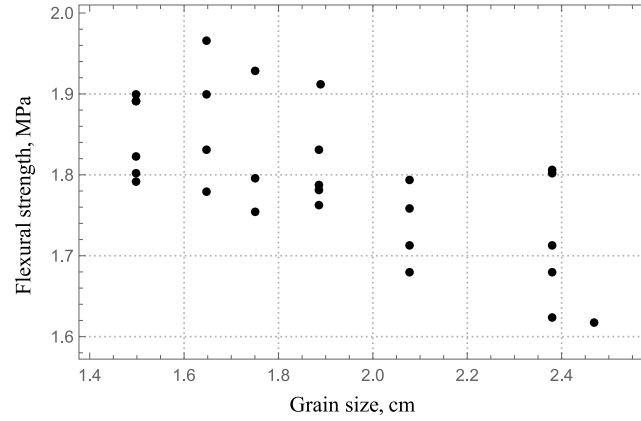


Figure 6.9: Relationship between the grain size and the flexural strength of the beam obtained from 28 simulations on beams with differing grain sizes.

sponding flexural strength is computed as

$$\sigma = \frac{3F(L - L_i)}{2bd^2},$$

where F is the load force at the fracture point, L is the length of the support (outer) span between the indenters, L_i is the length of the loading (inner) span, b is the width of the beam and d is the thickness of the beam. The results are summarized in Fig. 6.9 as a plot of flexural strength vs. grain size. As may be observed, the strength of the blocks increases with decreasing grain size, as expected from Hall-Petch relationship. The flexural strengths obtained from these simulations are in accord with several experimental results [24]–[26], however experimental measurements of the flexural strength of ice vary significantly [27] and the influence of scale effects is highly important [28]. Such effects will be explored further in future work.

6.4 Conclusion

The complexities of ice-structure interactions and the inherent variability in ice strength, coupled with the need for risk-based approaches in engineering design, result in a continued strong emphasis on empirical approaches in many branches of ice mechanics. Nevertheless, numerical modelling is an important tool for studying certain aspects of ice behavior, and the importance of numerical modelling will undoubtedly increase in the future as newer methods which capture the unique processes that take place in ice continue to be developed. In particular, detailed accurate modelling of ice fracture would be highly valuable in assessing load limiting processes and potential strategies for improving design.

In nature, deformation, fracture and motion of ice are affected by numerous factors, making the simulation of full-scale processes extremely challenging. The work presented here, although employed for the specialized case of polycrystalline ice under flexural loading, provides an improved technique which can be extended to model the behavior of less homogeneous ice under more general conditions. Limitations of the approach at this stage of development include the absence of processes associated with crushing; specifically microcracking, dynamic recrystallization and local pressure melting. Neither does the model account for viscous deformation, intragranular fracture and variabilities in grain-grain cohesion. However, it does capture grain size dependence, accumulation of crack-related damage and well represents experimental force-time curves. It also captures discrete fracture events, fragmentation and allows for the integration of more complex models of ice continuum behaviour in future work. Being a

new and flexible model, the PPR technique provides an improved approach for modeling ice using cohesive zones.

References

- [1] W. Weeks, *On Sea Ice*. University of Alaska Press, 2010, ISBN: 9781602231016.
- [2] M. Mellor, “Mechanical Properties of Polycrystalline Ice,” en, in *Physics and Mechanics of Ice*, P. Tryde, Ed., DOI: 10.1007/978-3-642-81434-1_17, Berlin, Heidelberg: Springer Berlin Heidelberg, 1980, pp. 217–245, ISBN: 978-3-642-81436-5.
- [3] P. D. Barrette and I. J. Jordaan, “Healed cracks in iceberg ice,” *Journal of Glaciology*, vol. 48, no. 163, pp. 587–591, 2002.
- [4] R. Taylor and I. Jordaan, “Probabilistic fracture mechanics analysis of spalling during edge indentation in ice,” *Engineering Fracture Mechanics*, vol. 134, pp. 242–266, Jan. 2015, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2014.10.021.
- [5] T. Browne, R. Taylor, I. Jordaan, and A. Gurtner, “Small-scale ice indentation tests with variable structural compliance,” *Cold Regions Science and Technology*, vol. 88, pp. 2–9, Apr. 2013, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2012.12.006.
- [6] K. B. Habib, R. S. Taylor, S. Bruneau, and I. J. Jordaan, “Experimental Study of Dynamics During Crushing of Freshwater Truncated Conical Ice Specimens,” ASME, May 2015, V008T07A014, ISBN: 978-0-7918-5656-7. DOI: 10.1115/OMAE2015-41904.

- [7] B. J. O'Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, "Experimental investigation of oscillation of loads in ice high-pressure zones, part 1: Single indenter system," *Cold Regions Science and Technology*, vol. 124, pp. 25–39, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.005.
- [8] B. J. O'Rourke, I. J. Jordaan, R. S. Taylor, and A. Gurtner, "Experimental investigation of oscillation of loads in ice high-pressure zones, part 2: Double indenter system — Coupling and synchronization of high-pressure zones," *Cold Regions Science and Technology*, vol. 124, pp. 11–24, Apr. 2016, ISSN: 0165232X. DOI: 10.1016/j.coldregions.2015.12.002.
- [9] P. Birajdar, R. Taylor, K. Habib, and R. Hossain, "Analysis of Medium-Scale Laboratory Tests on Ice Crushing Dynamics," Offshore Technology Conference, 2016. DOI: 10.4043/27482-MS.
- [10] P. Birajdar, R. Taylor, and R. Hossain, "Analysis of the Effect of Structural Compliance During Medium-Scale Laboratory Tests on Ice Crushing Dynamics," San Francisco, California, USA, Jun. 2017.
- [11] R. Taylor, T. Browne, I. Jordaan, and A. Gurtner, "Fracture and Damage During Dynamic Interactions Between Ice and Compliant Structures at Laboratory Scale," ASME, Jun. 2013, V006T07A024, ISBN: 978-0-7918-5540-9. DOI: 10.1115/OMAE2013-11070.

- [12] R. Lee and E. Schulson, “The Strength and Ductility of Ice Under Tension,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 110, no. 2, p. 187, 1988, ISSN: 08927219. DOI: 10.1115/1.3257049.
- [13] E. Schulson, “The brittle compressive fracture of ice,” *Acta Metallurgica et Materialia*, vol. 38, no. 10, pp. 1963–1976, Oct. 1990, ISSN: 09567151. DOI: 10.1016/0956-7151(90)90308-4.
- [14] R. Gagnon and P. Gammon, “Characterization and flexural strength of iceberg and glacier ice,” *Journal of Glaciology*, vol. 41, no. 137, pp. 103–111, 1995.
- [15] P. Moore, I. Jordaan, and R. Taylor, “Explicit Finite Element Analysis Of Compressive Ice Failure Using Damage Mechanics,” 2013.
- [16] K. Kolari, “A complete three-dimensional continuum model of wing-crack growth in granular brittle solids,” *International Journal of Solids and Structures*, vol. 115, pp. 27–42, 2017.
- [17] W. Lu, R. Lubbad, and S. Løset, “Simulating ice-sloping structure interactions with the cohesive element method,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 136, no. 3, p. 031 501, 2014.
- [18] F. Wang, Z.-J. Zou, L. Zhou, Y.-Z. Ren, and S.-Q. Wang, “A simulation study on the interaction between sloping marine structure and level ice based on cohesive element model,” *Cold Regions Science and Technology*, vol. 149, pp. 1–15, 2018.

- [19] I. Gribanov, R. Taylor, and R. Sarracino, “Cohesive zone micromechanical model for compressive and tensile failure of polycrystalline ice,” *Engineering Fracture Mechanics*, Apr. 2018, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2018.04.023.
- [20] I. Gribanov, R. Taylor, and R. Sarracino, “Parallel implementation of implicit finite element model with cohesive zones and collision response using CUDA: Implicit FEM CZs with CUDA Implementation,” *International Journal for Numerical Methods in Engineering*, Apr. 2018, ISSN: 00295981. DOI: 10.1002/nme.5825.
- [21] K. Park, G. H. Paulino, and J. R. Roesler, “A unified potential-based cohesive model of mixed-mode fracture,” *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 6, pp. 891–908, Jun. 2009, ISSN: 0022-5096. DOI: 10.1016/j.jmps.2008.10.003.
- [22] D. W. Spring, O. Giraldo-Londoño, and G. H. Paulino, “A study on the thermodynamic consistency of the Park–Paulino–Roesler (PPR) cohesive fracture model,” *Mechanics Research Communications*, vol. 78, pp. 100–109, Dec. 2016, ISSN: 00936413. DOI: 10.1016/j.mechrescom.2016.05.006.
- [23] R. Quey, P. Dawson, and F. Barbe, “Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 17–20, pp. 1729–1745, Apr. 2011, ISSN: 0045-7825. DOI: 10.1016/j.cma.2011.01.002.
- [24] G. W. Timco and R. Frederking, “Flexural strength and fracture toughness of sea ice,” *Cold Regions Science and Technology*, vol. 8, no. 1, pp. 35–41, 1983.

- [25] J.-C. Tatinclaux and C. Y. Wu, “On flexural strength of freshwater and saline ice,” in *Applied Techniques for Cold Environments*, ASCE, 1978, pp. 295–306.
- [26] B. Parsons, M. Lal, F. Williams, J. Dempsey, J. Snellen, J. Everard, T. Slade, and J. Williams, “The influence of beam size on the flexural strength of sea ice, freshwater ice and iceberg ice,” *Philosophical Magazine A*, vol. 66, no. 6, pp. 1017–1036, 1992.
- [27] L. Lainey and R. Tinawi, “The mechanical properties of sea ice—a compilation of available data,” *Canadian Journal of Civil Engineering*, vol. 11, no. 4, pp. 884–923, 1984.
- [28] M. Aly, R. S. Taylor, E. Bailey, and I. Turbull, “Scale effect in freshwater ice flexural strength,” in *Proceedings of OMAE 2018*, 2018.

7 | Investigation of Mixed Mode Fracture of L-shaped Sea Ice Beams

Preface

I am the primary author of this work, which was presented at the 25th International Conference on Port and Ocean Engineering under Arctic Conditions (POAC 2019). The co-authors are Aleksey Marchenko, Andrii Murdza, Rocky Taylor and Robert Sarracino. Marchenko and Murdza conducted extensive fieldwork, the results of which were published in 2017-2018. They also performed an analysis of stress distributions using the linear elastic model. Rocky Taylor suggested conducting an in-depth analysis of their data using cohesive zone model, which accounts for damage and fracture processes. Simulations allowed to predict the breaking force of the ice beams depending on the dimensions of the beam. While working on this project it became clear that the beam breaking force can be expressed as a function of the beam dimensions and the properties of ice. I suggested the first version of such formula, which was later improved by Aleksey Marchenko. The analytical formulae required verification, which I performed by comparing its prediction to the FEM results. Andrii Murdza and Robert Sarracino reviewed the manuscript and suggested several edits and clarifications.

Abstract

Modeling the fracture behavior of ice during mixed-mode loading is difficult. Historically, fracture testing in the field has focused primarily on experimental conditions such as flexural or tensile loading, which results in failure dominated by a single mode. Between 2014 and 2016, nine field experiments were performed on L-shaped cantilever beams where the deformation was a combination of bending and torsion. Two distinct fracture modes were recorded. In four of the nine cases, the failure occurred at the free end of the beam due to bending, and in five remaining cases, the failure occurred at the root of the beam due to torsion. Fracture paths are difficult to model under such conditions since they are influenced not only by geometry and loading conditions but also are affected by the variability of the material, including natural flaws present in the ice. In the current study, a cohesive zone model is applied to simulate the dynamic fracture processes in L-shaped beams. Evolution of the stress distribution on the surface of the beam is modeled for the duration of the loading process, showing how it changes with progressive accumulation of damage in the material, as well as the development of cracks. An analytical model is proposed for estimating the breaking force based on the dimensions of the beam, flexural and tensile strengths of the material. Dimensionless constants in the model are determined as the best fit for the simulation results. Finally, the experimental data obtained from the 2014-2016 tests are re-evaluated to infer the flexural and tensile strength of ice based on the proposed analytical model.

7.1 Introduction

Experimental measurements of mechanical properties of ice are performed quite regularly. Standardized testing methods are described by Schwarz et al. [1], and, more recently, by International Towing Tank Conference (ITTC) Recommended Procedures for testing ice properties. While there are several techniques for measuring flexural and compressive strengths, the only recommended method for measuring shear strength is the punch through test in which a hole is punched in a sheet of ice. An alternative way of investigating the shear strength was suggested by Murdza et al. [2], whose method is similar to in-situ plain beam loading, but instead uses an L-shaped cantilever beam. In such test, a pure bend is mixed with a torsional load, resulting in a “complex bend” – a superposition of bending (tensile) and plane (shear) stresses. Such loads are common in natural motion and breakup of ice, where the failure criterion cannot be formulated in terms of flexural strength alone.

Nine experiments on L-shaped cantilever beams were conducted between November 2014 and March 2016. The following information was recorded for each test:

- dimensions of the beam
- loading direction: upward or downward
- type of ice: sea or freshwater
- temperature and average salinity of ice

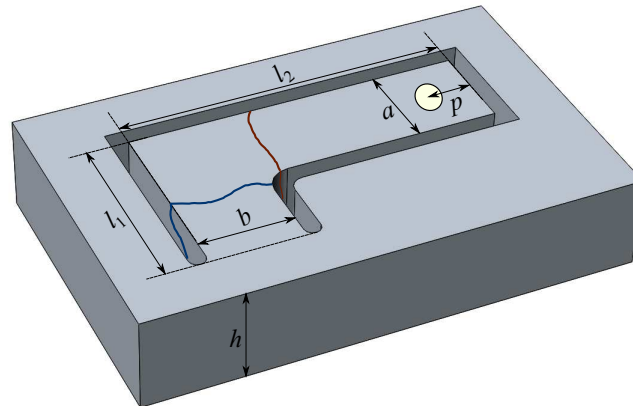


Figure 7.1: Dimensions of the L-shaped beam, indenter application point and two crack paths, which correspond to the free end fracture (shown in red color) and the fixed end fracture (shown in blue).

- force-time and displacement-time curves
- maximal indentation force and the corresponding indenter displacement
- angle and shape of the crack

Additionally, finite element analysis was performed to estimate the distribution of stress on the surface of the beam. For the constitutive model of ice, isotropic linear elastic material was selected. By inspecting the first and third principal stresses areas of stress concentrations were located. However, those areas did not coincide with the experimentally observed fracture paths.

While none of the fracture paths were the same in the experiment, their locations can be classified into two groups: (1) at the free end and (2) at the fixed end (Figure 7.1). The experiments and subsequent simulations showed that “in-between” option is unlikely, and fractures usually

occur on either side of the bend. Failures at the free end are analogous to the experiments with plain cantilever beams, failing in pure flexure. In such a case, even if the shear stresses were present, the maximum recorded load corresponds to the sample strength in transverse bending. On the other hand, failures at the fixed end correspond to torsional failure, where shear strength plays the main role.

The present work attempts to re-evaluate the obtained experimental data with the aid of the recently implemented cohesive zone model for ice fracture [3] and calculate the corresponding flexural and tensile strengths. Additionally, the analytical solution is suggested to determine whether the sample will break by flexure or torsion. The analytical approach may be helpful for quick estimates of the breaking force and type of failure, which depends on the dimensions of the beam and mechanical properties of the material. For brevity, beam dimensions will be presented as ordered list of the form $\{l_1, l_2, a, b, h\}$ with values expressed in meters.

7.2 Cohesive zone modeling

The non-linear behavior of ice was summarized by Sanderson [4]. For the processes with large deformations and fragmentation, researchers still struggle to formulate a suitable constitutive model of ice. Various approaches to modeling ice fracture include continuum models for material damage [5], wing crack growth [6], analysis based on the specific energy of fracture [7], probabilistic fracture mechanics [8] and cohesive zone model [3]. The latter was successfully applied to analyze uniaxial compressive and tensile tests of ice samples. The Park-Paulino-

Roesler formulation [9], [10] has eight parameters that determine traction-separation curves of cohesive zones in normal and tangential directions. This flexibility allows to model damage accumulation and fracture in a wide range of materials including ice.

Establishing the parameters of a CZ model is a non-trivial task. Experimental measurements of ice strength usually provide a single quantity, such as the maximum indentation force before failure. This quantity can be converted to either flexural, tensile or shear strength, depending on the type of the experiment. On the other hand, the CZ model requires such values as specific fracture energies, normal and cohesive strengths σ_{max} and τ_{max} , shape coefficients of traction-separation curves and initial slopes of these curves. Some of the required values can be established based on small-scale laboratory tests, but the main difficulty lies in determining the normal and tangential strengths of cohesive zones σ_{max} and τ_{max} based on the limited number of experimental results.

The first simulation is performed on a beam with dimensions $\{1.05, 1.95, 0.55, 0.55, 0.56\}$. A tetrahedral mesh is created with *gmsh* library, with the element count of 8582 and cohesive zone count of 8339. In this and subsequent tests, the radius of rounding on the inner side of the beam is set to 0.2m (twice the size of the gap). Cohesive zones are inserted between the elements in the area where the fracture is expected (Figure 7.2). Parameters of the material and cohesive zones are listed in Table 7.1. An attempt is made to use simulation parameters that result in breaking forces similar to the ones observed in the field experiments. The goal for performing the simulation is to observe the evolution of stress distribution and to generate a set of data points for differing beam dimensions, keeping the material parameters fixed.

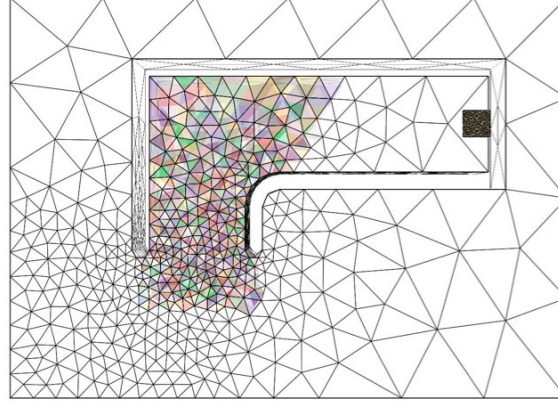


Figure 7.2: Tetrahedral mesh for FE simulation. Cohesive zones are inserted between the highlighted elements. Indenter measuring 15×15 cm applies force to the free end of the beam.

Table 7.1: Parameters of cohesive zones and elastic elements.

Notation	Description	Value
σ_{max}	Normal cohesive strength	120 kPa
τ_{max}	Tangential cohesive strength	140 kPa
ϕ_n, ϕ_t	Mode I and II fracture energies	3 Jm^{-2}
α, β	Shape of traction-separation curves (determine brittleness)	4
λ_n, λ_t	Non-dimensional slope indicators in PPR model	0.015
E	Young's modulus	5 GPa
ν	Poisson's ratio	0.3
ρ	Density of the material	916.2 kg/m^3

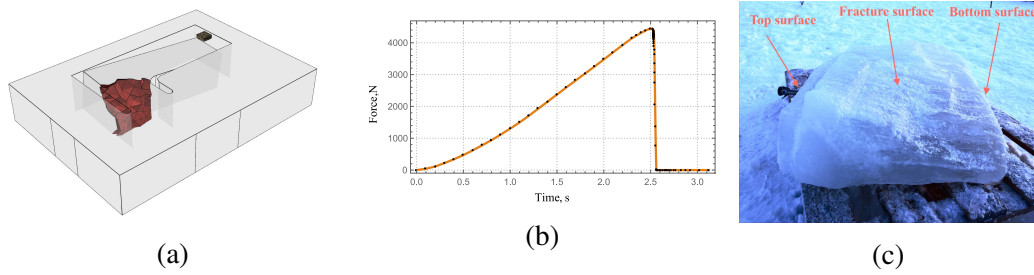


Figure 7.3: (a) Resulting fracture surface. (b) Force on the indenter vs. time. (c) A typical fracture surface obtained in the field test.

The force is applied at the free end of the beam via rectangular indenter moving downwards at 1.7 mm/s (Figure 7.3a). The graph of force on the indenter versus time is shown in Figure 7.3b. The initial time step is set to 0.1 s, and the adaptive algorithm decreases it to 0.1/2048 when needed. For brittle cylinders under torsional load the fracture usually has a helicoidal shape [11]; in the current simulation, the fracture surface is slightly curved (Figure 7.3a). The fracture surface in the simulation compares favorably with fracture surfaces observed in the field experiments. In one of the field tests the root part of the beam, which remained attached to the ice cover after the beam failure, was cut off with a hand saw and placed on the wooden pallet in the way that the cut surface was on the bottom, in contact with the pallet, and the fracture surface was on the top (Figure 7.3c). The fracture surface is curved and inclined; the distance from the beam root to the fracture surface is smaller on the bottom of the beam; the distance from the beam root to the fracture surface on the bottom of the beam is smaller from the side of a shorter lever or at the rounding B in Figure 4a in Murdza et al. [2]. Thus, the fact that the fracture surfaces obtained in both simulation and field experiment are almost identical is a good sign of the accuracy of the simulation.

Evolution of the distribution of first principal stress is shown in Figure 7.4. At the initial stages

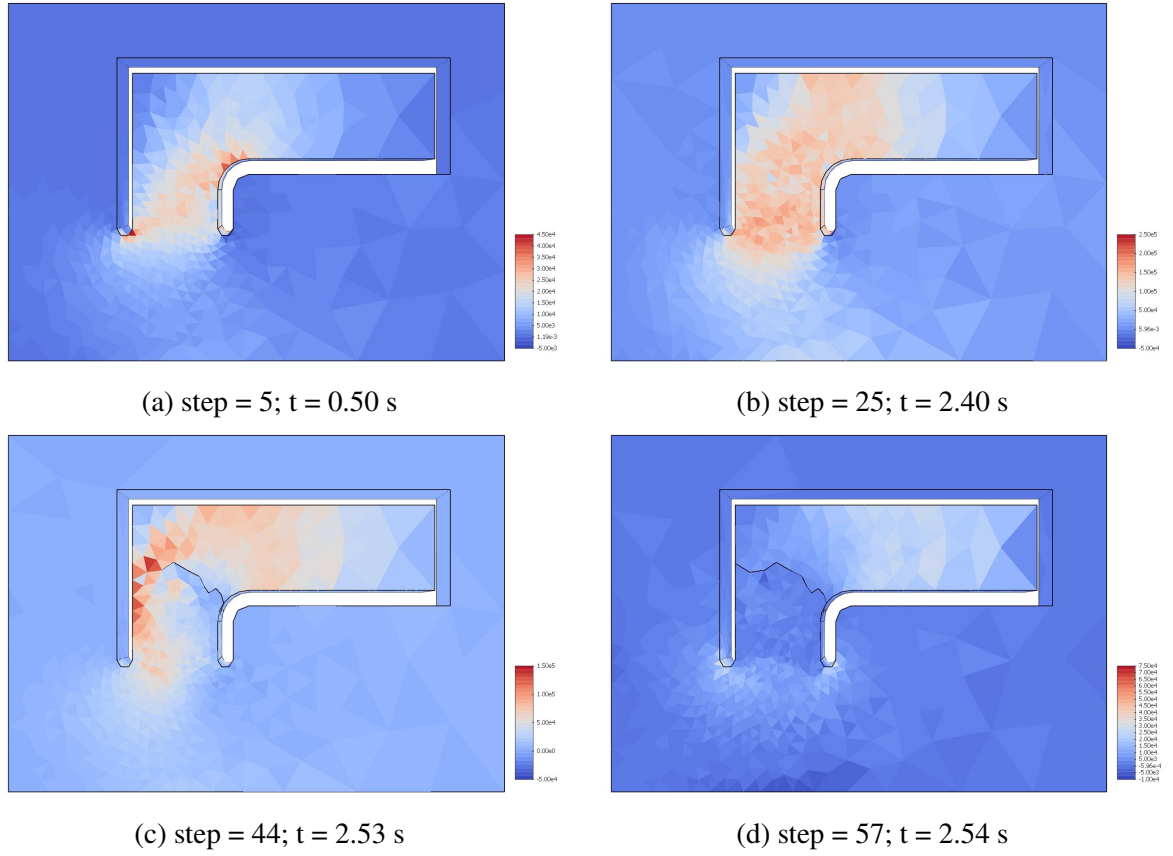


Figure 7.4: Distribution of the first principal stress over the top surface of the beam at various time steps. The units on the legends are Pascals.

of loading, two areas of high stress concentration develop (Figure 7.4a), where the values reach 66 kPa. Such stress concentration is not sufficient to damage cohesive zones, whose strength is about twice higher. In the initial stage, the distribution of stress on the surface is similar to the linear elastic case.

First damaged cohesive zones appear on step 16 in the lower-left corner of the beam. At step 25 about 3% of cohesive zones become damaged (Figure 7.5a), and stress distribution is visibly different from the linear case (Figure 7.4b). Between steps 18 and 30, the maximum value of the first principal stress stays around 200 kPa and starts to drop when the fracture is

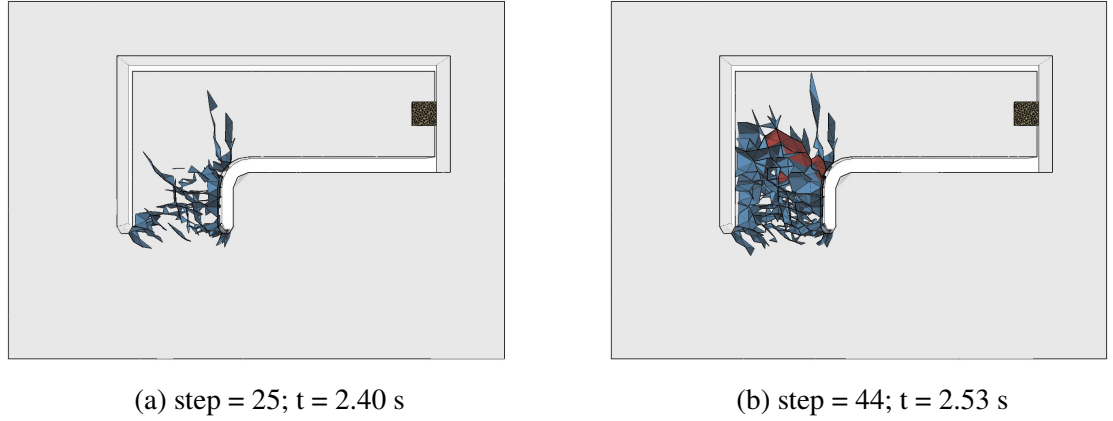


Figure 7.5: Accumulation of damaged cohesive zones, shown in blue color, and propagating crack, shown in red.

initiated at step 30; a reduction of stress concentration is then seen near the propagating crack (Figures 7.4c, 7.5b). After the propagation is complete, the internal damage results in residual internal stresses (Figure 7.4b).

7.3 Analytical approach

If the failure occurs at the free end, the process can be viewed as plain cantilever beam bending. The breaking force, which is determined by the flexural strength, can be approximated via equation for the elastic beam:

$$F = \frac{\sigma_f w h^2}{6l}, \quad (7.1)$$

where σ_f is the flexural strength, l is the length, w is the width and h is the height of the beam. Expression (7.1) is derived for the fixed-end plain beam and is not expected to yield precise results for the L-shaped beam, where the fracture is not perpendicular to the main

axis and torsional deformation leads to non-symmetric stress distribution with areas of high stress concentration. This effect is influenced by the width of the fixed-end fragment, which expression (7.1) does not include. To account for the approximate application of formula (7.1), a dimensionless coefficient k is introduced. The length l is taken to be $(l_2 - b - p)$, where l_2 is the length of the free end, b is the width of the fixed end, and p is the point of application of the indenter (Figure 7.1). Substituting the length l into expression (7.1) yields the approximation for the breaking force

$$F_p = \frac{\sigma_f w h^2}{6(l_2 - b - p)}. \quad (7.2)$$

In this and subsequent calculations, the indenter application point p is 0.075 m. By fitting equation (7.2) to the results of finite element simulations, the value of the dimensionless coefficient was found to be $k = 0.831$. It shows that the breaking force of the L-shaped beam is about 83.1% of the breaking force of a corresponding plain cantilever beam.

A different approach is taken when the fracture takes place at the fixed end. In such a case, the loading process is a combination of torsion and bending, and both components influence the breaking force. The bending component can be approximated as

$$F = \alpha \frac{\sigma_b w h^2}{6l_1}, \quad (7.3)$$

where α is a dimensionless constant, σ_b is the bending (flexural) stress and l_1 is the length of the fixed end. To estimate the torsional component, the beam is viewed as a solid rectangular cross-section to which a twisting moment is applied. The free end of the beam acts as a lever

that applies torque $T = F(l_2 - p)/\beta$, where β is another dimensionless constant. Maximum shear stress on the surface of rectangular solid under torsional load is expressed as [12]:

$$\sigma_t = \frac{3T}{bh^2} \left[1 + 0.6095 \frac{h}{b} + 0.8865 \left(\frac{h}{b} \right)^2 - 1.8023 \left(\frac{h}{b} \right)^3 + 0.9100 \left(\frac{h}{b} \right)^4 \right]. \quad (7.4)$$

Formula (7.4) is derived with the condition $b \geq h$, where b is the width and h is the height of the beam. For the cases where $b < h$, parameters b and h must be swapped.

Stress tensor at the surface of the fixed end is written in the form

$$\sigma = \begin{pmatrix} \sigma_b & \sigma_t \\ \sigma_t & 0 \end{pmatrix}, \quad (7.5)$$

whose principal values are equal to

$$\sigma_1 = \frac{\sigma_b}{2} + \sqrt{\left(\frac{\sigma_b}{2} \right)^2 + \sigma_t^2}, \quad \sigma_2 = \frac{\sigma_b}{2} - \sqrt{\left(\frac{\sigma_b}{2} \right)^2 + \sigma_t^2}. \quad (7.6)$$

The failure criterion is formulated as

$$\sigma_1 = \sigma_{ten}, \quad (7.7)$$

where σ_{ten} is the tensile strength on the surface, which is equal to flexural strength when

$\sigma_t = 0$. The first principal direction is specified by equation

$$y = \frac{\sigma_1 - \sigma_b}{\sigma_t} x, \quad (7.8)$$

where the axes x and y are directed along the levers l_1 and l_2 . The failure surface is perpendicular to the first principal direction and starts from the beam root because of the stress concentration. If $\sigma_t = 0$ then the failure surface is parallel to y -axis and corresponds to the crack in the beam root in bending failure. Substituting expressions (7.3) and (7.4) into (7.7) yields

$$F_l = \begin{cases} \frac{\alpha \sigma_{ten} b h^2}{3 l_1 \left(1 + \sqrt{1 + \left(\frac{2 \alpha \gamma (l_2 - p)}{\beta l_1} \right)^2} \right)}, & b \geq h \\ \frac{\alpha \sigma_{ten} b h^2}{3 l_1 \left(1 + \sqrt{1 + \left(\frac{2 \alpha h \tilde{\gamma} (l_2 - p)}{\beta l_1 b} \right)^2} \right)}, & b < h \end{cases} \quad (7.9)$$

Where

$$\gamma = 1 + 0.6095 \frac{h}{b} + 0.8865 \left(\frac{h}{b} \right)^2 - 1.8023 \left(\frac{h}{b} \right)^3 + 0.9100 \left(\frac{h}{b} \right)^4,$$

$$\tilde{\gamma} = 1 + 0.6095 \frac{b}{h} + 0.8865 \left(\frac{b}{h} \right)^2 - 1.8023 \left(\frac{b}{h} \right)^3 + 0.9100 \left(\frac{b}{h} \right)^4.$$

Expressions (7.2) and (7.9) agree very well with the results of the finite element simulation. That is, for the material with known flexural strength σ_f and tensile strength σ_{ten} , the breaking force of the L-shaped beam can be predicted. Expression (7.2) predicts the breaking force F_p in the flexural mode, and expression (7.9) predicts the mixed-mode failure F_l . The weakest of the two forces determines the actual location of the crack.

Table 7.2: Series of parametric tests.

Varying parameter	Parameter name	Parameter range, [m]	Beam dimensions $\{l_1, l_2, a, b, h\}$	Number of tests
l_1	Length of the fixed end	0.9–1.3	$\{-, 1.7, 0.55, 0.55, 0.56\}$	20
l_1	Length of the fixed end	0.9–1.8	$\{-, 1.6, 0.5, 0.55, 0.5\}$	20
l_2	Length of the free end	1.3–3.0	$\{1.05, -, 0.55, 0.55, 0.56\}$	20
a	Width of the free end	0.3–0.6	$\{1.05, 1.95, -, 0.55, 0.56\}$	20
b	Width of the fixed end	0.15–0.85	$\{1.05, 1.95, 0.55, -, 0.56\}$	33
h	Thickness	0.15–0.90	$\{1.05, 1.95, 0.55, 0.55, -\}$	30

7.3.1 Comparison of the analytical solution with FEM parametric tests

Computer simulations allow to test the strengths of beams of varying dimensions quickly. A set of parametric FEM studies was conducted and summarized in Table 7.2. Material parameters were the same for all tests (Table 7.1). Results of the parametric studies are used to establish parameters for the analytical expressions.

To determine the coefficients α and β , the best fit is found for expression (7.9) based on parametric FEM studies (Table 7.2). Parametric tests are separated into two fracture modes with the first group containing fractures at the free end (Figure 7.6) and the second group containing fractures at the fixed end (Figure 7.3a). The latter group is used for fitting coefficients α and β in expression (7.9). The obtained values of the coefficients are $\alpha = 2.758$ and $\beta = 4.588$. By performing a uniaxial tension simulation, the tensile strength of the simulated material is

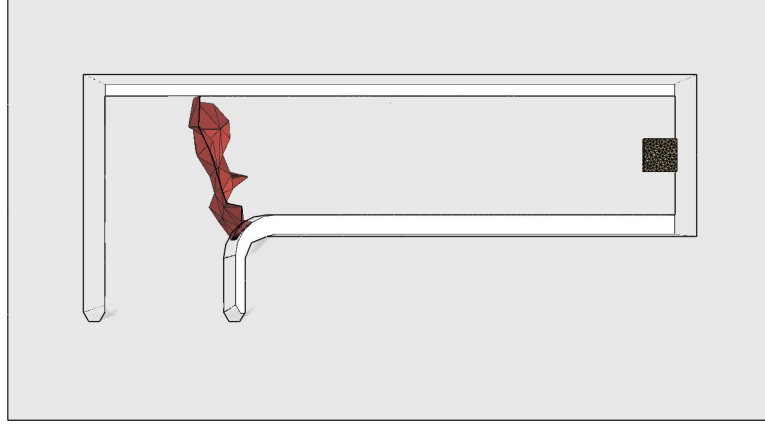


Figure 7.6: Simulated fracture at the free end of the beam.

determined to be $\sigma_{ten} = 138501$ Pa.

The first parametric study varies the parameter l_1 for beams with dimensions $\{-, 1.7, 0.55, 0.55, 0.56\}$ and $\{-, 1.6, 0.5, 0.55, 0.5\}$. Equation (7.9) accurately predicts the decrease of the breaking force with the increasing length l_1 . Blue lines on Figure 7.7 correspond to the breaking force for flexural failure (at the free end), but the actual fracture takes place at the fixed end with lower force. Results of the finite element simulations show some variability due to the randomness in finite element meshing, but they follow the analytical prediction.

The next parametric study varies the length of the free end l_2 from 1.3 to 3.0 meters, with 20 gradations between (Figure 7.8a). The beam dimensions are $\{1.05, -, 0.55, 0.55, 0.56\}$. Up until the length of 2.19m, the beams break at the fixed end, whereas longer beams break at the free end. The point where the transition occurs is captured by the analytical model as the intersection of equations (7.2) and (7.9).

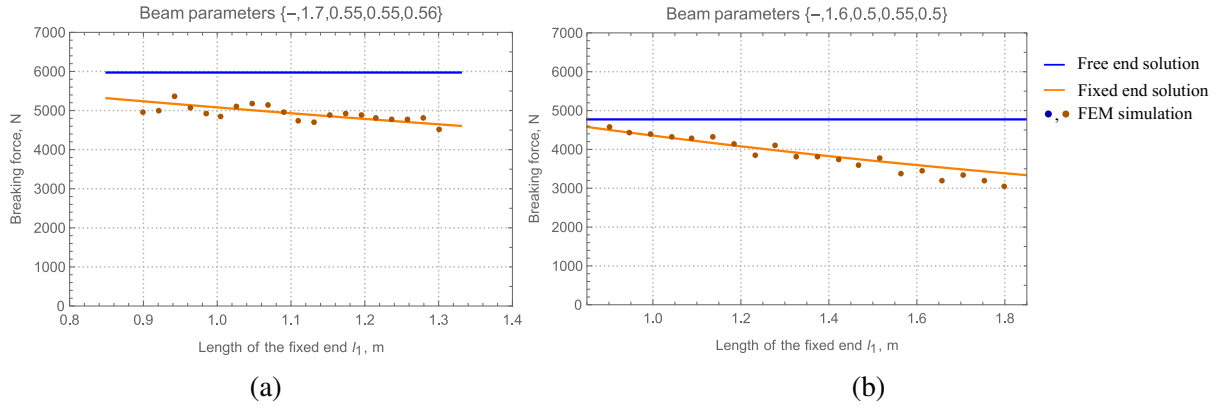


Figure 7.7: Parametric study for varying length of the fixed end.

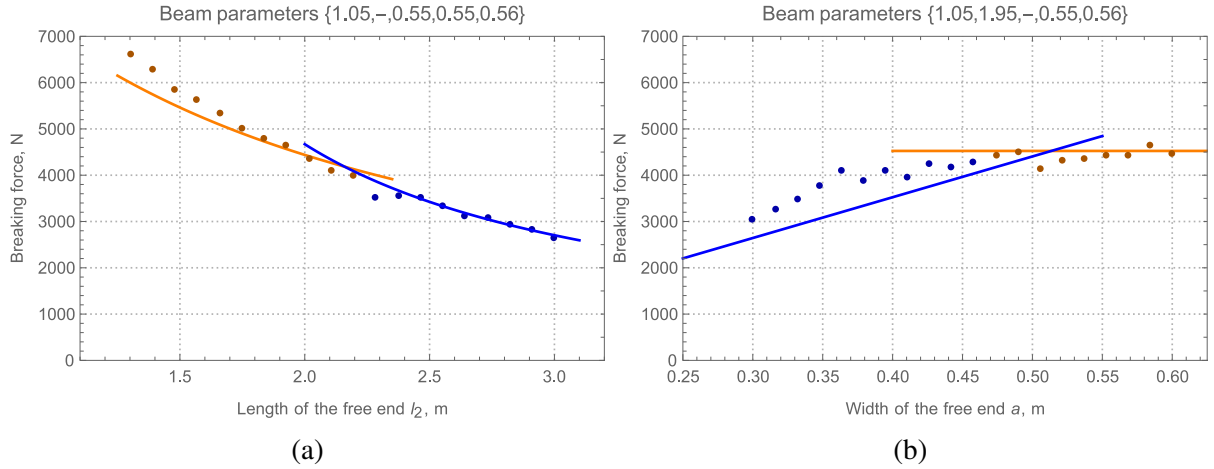


Figure 7.8: Parametric study for varying (a) length of the free end, (b) width of the free end.

Parametric study of the width of the free end shows that wider beams result in higher breaking forces, but only in the region where the fracture occurs at the free end. After reaching a certain width, the wide free end becomes strong enough to sustain the load, and the fracture mode shifts to the fixed end, as illustrated in Figure 7.8b.

Two additional studies were conducted to verify the applicability of expression (7.9). Both studies expose the non-linear dependence between the breaking force and the beam dimensions. When the fixed end is narrow (Figure 7.9a) it fractures easily. However, the breaking

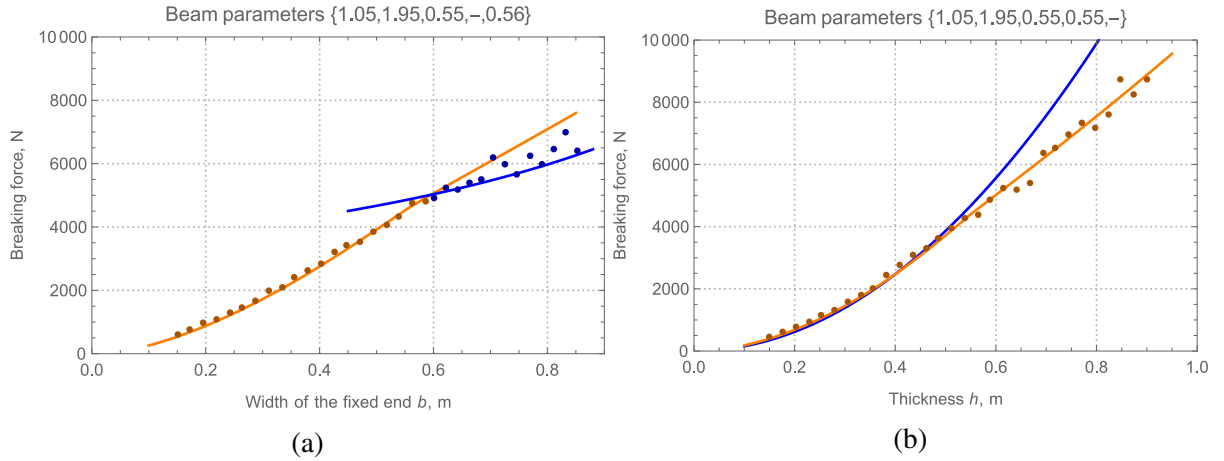


Figure 7.9: Parametric tests for varying (a) width of the fixed end, (b) thickness of the sheet.

force grows non-linearly with increasing width. After a certain width is reached, the fixed end becomes stronger than the free end and the fracture mode changes, which is reflected in Figure 9a. Varying the thickness of the sheet also has a non-linear relationship with the breaking force (Figure 7.9b). In both cases, the proposed analytical expressions (7.2) and (7.9) capture the complex relationship and give accurate predictions.

7.3.2 Application of the analytical model to the field test data

The field tests performed by Murdza et al. [2] recorded beam dimensions alongside with the breaking forces. By applying the proposed analytical model, additional information about the material can be inferred. When the fracture occurs at the free end, the flexural strength can be determined via equation (7.2). For the tests where fractures happened at the fixed end, tensile strength can be determined via equation (7.9). This data is calculated and summarized in Table 7.3.

Table 7.3: Calculated flexural/shear strength for experimentally obtained forces.

No.	Ice type	l_1 [m]	l_2 [m]	a [m]	b [m]	h [m]	F_{max} [N]	σ_f [kPa]	σ_{ten} [kPa]
1	sea	1.05	1.95	0.55	0.55	0.56	6267	–	194
2	sea	1.2	2.1	0.6	0.6	0.53	3881	–	130
3	sea	1.2	2.08	0.6	0.65	0.6	4506	–	109
4	fresh	0.75	1.1	0.42	0.36	0.36	2559	–	173
5	fresh	0.68	1.15	0.4	0.44	0.36	3528	285	–
6	fresh	–	2.55	0.85	0.85	0.28	1436	231	–
7	sea	1.3	2	0.62	0.65	0.59	6897	269	–
8	sea	1.15	1.45	0.55	0.6	0.59	7064	189	–
9	sea	1.57	1.22	0.9	0.57	0.59	6142	–	147

7.4 Discussion and conclusions

The tests presented in Table 3 were performed at three different locations. Even the samples 1-3 that come from the same site show high variability in tensile strengths, ranging between 109kPa and 194 kPa, which suggests that the tested ice contained random inclusions that influence test results. Marchenko and Sakharov [13] performed laser scanning of the fractured surfaces and discovered the so-called “cheese ice,” whose surface contained cylindrical holes. Therefore, non-homogeneous natural ice can differ in its mechanical properties even at the same test site.

Performing large-scale and medium-scale tests in the field is a labor-intensive task; and usually, only a small number of data points are collected. Computer simulations provide more insight about the fracture processes, allowing to run parametric studies. In the case of the L-shaped

beam, it may be possible to predict, to some extent, whether the beam would break at the fixed or free end. This approach may be useful, for example, if the goal is to perform fixed-end torsional fractures only.

The proposed analytical approach has several simplifications. It does not account for the bending geometry of the beam and does not consider the progressive accumulation of damage. In comparison with the finite element modeling, however, the results are much faster to obtain, still providing good accuracy.

The finite element simulation accounts for the dynamics of the process, gradual accumulation of damage and redistribution of the stress during the loading process. The modeled loading curves, the locations of the cracks and their shapes generally agree with the experimental results. To some extent, simulation results depend on the mesh topology, and the predicted breaking force may be slightly different for different mesh geometries. Another drawback of the simulation is that it requires significant time to compute, i.e. about 20 minutes per test. It is formulated in terms of normal and tangential strengths of cohesive zones, which are not measured directly. Analytical expressions are easier to use and may be applied to estimate the force and type of fracture quickly.

References

- [1] J. Schwarz, R. Frederking, V. Gavrillo, I. Petrov, K.-I. Hirayama, M. Mellor, P. Tryde, and K. Vaudrey, “Standardized testing methods for measuring mechanical properties of ice,” en, *Cold Regions Science and Technology*, vol. 4, no. 3, pp. 245–253, Jul. 1981, ISSN: 0165232X. DOI: 10.1016/0165-232X(81)90007-0.
- [2] A. Murdza, A. Marchenko, A. Sakharov, P. Chistyakov, E. Karulin, and M. Karulina, “Test with L-shaped cantilever beam for complex shear and bending strength,” in *Proceedings of the 23rd IAHR International Symposium on Ice.*, 2016.
- [3] I. Griбанov, R. Taylor, and R. Sarracino, “Cohesive zone micromechanical model for compressive and tensile failure of polycrystalline ice,” *Engineering Fracture Mechanics*, Apr. 2018, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2018.04.023.
- [4] T. Sanderson, *Ice mechanics: risks to offshore structures*. London, UK ; Boston: Graham & Trotman, 1988, ISBN: 978-0-86010-785-9.
- [5] M. B. Kavanagh, “Time-dependent aspects of fracture in ice,” PhD thesis, Memorial University of Newfoundland, 2018.
- [6] K. Kolari, “A complete three-dimensional continuum model of wing-crack growth in granular brittle solids,” *International Journal of Solids and Structures*, vol. 115, pp. 27–42, 2017.

- [7] V. Tsuprik, A. Bekker, E. Pomnikov, and E. Ivolgin, “Experimental Researching of the Specific Energy Mechanical Fracture of Ice by Method of Uniaxial Compression of Samples,” in *24rd International Conference on Port and Ocean Engineering under Arctic Conditions*, Busan, Korea, 2017.
- [8] R. Taylor and I. Jordaan, “Probabilistic fracture mechanics analysis of spalling during edge indentation in ice,” *Engineering Fracture Mechanics*, vol. 134, pp. 242–266, 2015.
- [9] K. Park, G. H. Paulino, and J. R. Roesler, “A unified potential-based cohesive model of mixed-mode fracture,” *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 6, pp. 891–908, 2009.
- [10] K. Park and G. Paulino, “Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces,” in *Applied Mechanics Reviews*, vol. 64, no. 6, p. 060 802, Feb. 2013, ISSN: 0003-6900. DOI: 10.1115/1.4023110.
- [11] H. E. Davis, G. E. Troxell, and G. F. Hauck, “The testing of engineering materials,” 1982.
- [12] R. J. Roark, W. C. Young, R. G. Budynas, and A. M. Sadegh, *Roark’s formulas for stress and strain*, 8th ed. New York: McGraw-Hill, 2012, OCLC: ocn769455976, ISBN: 978-0-07-174247-4.
- [13] N. Marchenko and A. Sakharov, “Laser scanning in mechanical ice tests,” in *24th International Conference on Port and Ocean Engineering under Arctic Conditions*, Busan, Korea, 2017.

8 | Summary and Conclusion

The goal of the research described in this thesis is to extend the toolset available for modeling ice and to move beyond the established techniques of curve fitting, particle simulation, and linear elastic fracture mechanics. Nonlinear fracture behavior of ice is investigated with the aid of finite element and cohesive zone models. Ultimately, this research aims to accurately reproduce the fracture process of small- and medium-size samples of brittle polycrystalline ice and give insight into the details of the fracture process. Several examples are discussed, including cylindrical samples under uniaxial compression and tension, four-point beam bending tests and L-shaped beam bending tests. The contribution of the current research is the validated numerical framework for the investigation of fracture, which contains several innovative techniques that set a baseline for future analyses. This chapter summarizes the thesis and its contributions and provides suggestions for future work.

8.1 Summary

The versatile behavior of natural ice and the difficulties of modeling are explained in the first chapter. Established methods of simulation are discussed, as well as the circumstances when they fail. A brief overview of naturally occurring ice-related events is given, and several

experimental techniques are introduced, which come from the area of computer graphics but may be applicable for engineering applications. The chapter establishes a common basis for the research presented in this thesis and also hints at several perspective methods that are worth exploring. The second part of Chapter 1 gives a brief review of continuum mechanics, finite element method, cohesive zone model, and collision detection and response. Chapter 2 contains the literature review on ice mechanics.

To conduct this research, a set of tools has been developed and compiled. These tools are described in detail in Chapters 3 and 4. Chapter 3 introduces the PPR cohesive zone formulation and focuses on its application to the uniaxial compression and tension experiments on the small-sized samples of polycrystalline solids. Attention is given to the generation of the polycrystalline geometry in a way that resembles the laboratory-made samples. The method of numerical simulation is described briefly; a large portion of the chapter is dedicated to the comparison of the numerical results with the experiments.

Chapter 4 explores the parallel implementation of the model, giving particular attention to the role of GPU computation. It outlines several technical challenges related to the assembly of the sparse stiffness matrix and proposes efficient ways to address these challenges. The chapter also highlights how different formulations of unloading/reloading relations can influence the computation in the case of implicit integration. Examples of brittle and ductile materials are given to illustrate the modeling framework, and simple benchmarking is done to show which parts of the simulation are computationally expensive.

During the fracture simulation, an initially intact object becomes fragmented, and the resulting pieces interact through contacts. When implementing a collision response scheme, a necessity arose for evaluating the distance between a point and a triangle in 3D, as well as for computing the first and second derivatives of that distance with respect to the coordinates of the involved points. It was found that no open-source algorithms were available for this purpose, so a new algorithm was developed. Implementations in C#, C, and Mathematica were written, and their accuracies were evaluated. An open-source C version is made available. Chapter 5 describes the derivations of the formulae and the details of the implementation and testing of this algorithm, which is an essential component of the work described in this thesis.

In Chapter 6 the proposed numerical method is applied to modeling the four-point bending of polycrystalline ice beams. Similar experiments are conducted regularly to investigate the flexural strength of ice in different regions and conditions. Computer simulation gives an insight into the fracture processes and confirms the relationship between the grain size and the flexural strength of the beam, i.e., Hall-Petch strengthening.

Chapter 7 investigates the fracture behavior of L-shaped sea ice beams. Previously conducted field tests were designed to evaluate the strength of ice under complex bending. In addition to recording the breaking force, the location and shape of the crack were also recorded. Results showed that the fracture took place at either the free end or the fixed end of the beam. Simulations were performed with the goal of establishing a relationship between the beam's dimensions and the location of the fracture. The results of 140 simulations were used to validate an analytical formula that predicts the breaking force and fracture location of the beam. The

obtained result may have a practical application in the future fieldwork on L-shaped beams.

The main contributions of this thesis are summarized as follows:

- The development and application of a new cohesive zone model for simulating the mechanics of ice failure has been completed. Studied cases include uniaxial loads on cylinders, four-point beam bending tests and L-shaped beam bending tests [1]–[3]. The effect of grain boundary strengthening was observed in numerical simulations and compared to the experimental data.
- A novel open-source parallel GPU implementation approach has been developed and made available for processing elements and CZs. The full source code is distributed with sample setups [4].
- A novel approach for numerical implementation has been developed and described, including a new procedure for initializing the sparse matrix, the collision scheme, and the memory layout in parallel portions of the code [4].
- The effect of different unloading/reloading relations on obtaining convergent solutions in the implicit scheme, which is of particular interest in mixed-mode scenarios, was investigated [4].
- An algorithm for evaluating the gradient and the Hessian of the point-triangle distance in 3D was developed to enable efficient computation of ice fragment contacts and has been implemented and freely distributed [5].

- An analytical expression has been formulated for the prediction of the breaking force of L-shaped sea ice beams and found to yield good agreement with numerical simulations and field tests results [3].

8.2 Discussion

This chapter would not be complete without looking into shortcomings that were encountered throughout the development of this project. Such a discussion would help other students and researchers to avoid pitfalls and focus on more fruitful routes.

From the beginning of the present project, efficient computation was a significant consideration. GPU was used to accelerate various portions of the computer code, such as the assembly of the linear system and even, at some point, finding the sums of the arrays. The code to perform these tasks was written in CUDA C++, which is a low-level programming language with manual memory management. GPU is most efficient when working with arrays of primitive types rather than the arrays of classes, and, as a result, the GPU code is often larger and more challenging to maintain. For small research projects, especially at the stage of development, cleaner high-level code is preferable to the highly optimized low-level code.

In this particular work, another shortcoming of the GPU acceleration is that the benefit of faster calculations is negated by the necessity of using a CPU-based linear solver. While parallel operations usually scale well, e.g., to trillions of particles in particle-based simulations,

the linear solver is only capable of handling several million unknown variables efficiently. Since the CPU-based solver is the bottleneck, optimizing the rest of the code provides only a moderate benefit. For the discussed simulation, a CPU-based implementation, written entirely in C#, was only 50% slower.

For research work whose topic is high-performance computing itself, parallelization may be a requirement. But for new model formulations, where efficiency is not the primary concern, using the less-efficient computation methods may save time for other work.

While the behavior of ice is a broad topic, numerical modeling is also a large area with many branches. Learning how to use a particular set of tools, e.g. MATLAB, ABAQUS, CUDA, Intel Math Kernel Library, takes time and effort. Researchers tend to use the tools they are already familiar with and develop their projects accordingly. This subjective consideration sometimes leads to using incorrect tools for a given task or to adjusting the goals of the project to match the available toolset. In this work, an extensive effort was spent on selecting the right tools for implementing the FEM computations, creating granular geometry and other tasks. Many third-party libraries were tested and ultimately rejected because they did not fit the needs of this project. Knowing the proper set of tools at the initial stage of work is definitely a time saver.

8.3 Recommendations for future work

During the realization of this project, several perspective ideas were formulated but never pursued due to their complexity or the lack of time. Some of them may have good potential for the simulation of several phenomena related to the natural behavior of ice. Three areas identified here as future work recommendations are given below.

The first recommendation is the application of the model for thin sheet fracture to the phenomenon of ice breakup on ponds and rivers. Natural ice covers often have a uniform thickness, primarily when formed in still water. These covers tend to break in flexure and tension, most commonly perpendicular to the plane of the sheets. Thin sheets are modeled in three dimensions, but their geometry is constrained to a curved surface. Such an approach can accurately represent the interaction of ice with bridge piers, for example. This concept may be viewed as a modification of the DEM approach, which is already used to model such events. The difference is in the fracture criterion and the representation of the geometry (mesh vs. particles). The mesh-based method may provide better accuracy because with the mesh-based approach the boundaries of the floes are defined explicitly.

The second recommendation is to address the viscous flow of ice, particularly under high confinement in the case of a high pressure zone. During indentation experiments, the ice softens and flows in the vicinity of the indenter. Sometimes the material flows slowly in the form of compressed powder or slush, whereas sometimes it is ejected rapidly and pulverized.

Such behaviors were traditionally modeled with particle-based methods and are difficult to handle with meshes. With the recent development of local mesh refinement and merging, viscous flow can be successfully modeled with a finite element method. This approach may allow better reproduction of the indentation processes associated with high-pressure zones.

The cohesive zone model that is extensively utilized in the present work uses the Park-Paulino-Roesler formulation, which is highly flexible and possesses many good qualities in comparison with other formulations. This formulation, however, does not take into account the strain rate dependence, which is known to exist in ice. A rate-dependent CZ formulation is needed to capture such effect for polycrystalline ice.

References

- [1] I. Griбанov, R. Taylor, and R. Sarracino, “Cohesive zone micromechanical model for compressive and tensile failure of polycrystalline ice,” *Engineering Fracture Mechanics*, Apr. 2018, ISSN: 00137944. DOI: 10.1016/j.engfracmech.2018.04.023.
- [2] I. Griбанov, R. Taylor, and R. Sarracino, “Application of cohesive zone model to the fracture process of freshwater polycrystalline ice under flexural loading,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 193, 2018, p. 012013.
- [3] I. Griбанov, A. Marchenko, A. Murdza, R. Taylor, and R. Sarracino, “Investigation of mixed mode fracture of l-shaped sea ice beams,” in *Proceedings of the 25th Inter-*

national Conference on Port and Ocean Engineering under Arctic Conditions POAC, 2019.

- [4] I. Griбанov, R. Taylor, and R. Sarracino, “Parallel implementation of implicit finite element model with cohesive zones and collision response using CUDA: Implicit FEM CZs with CUDA Implementation,” *International Journal for Numerical Methods in Engineering*, Apr. 2018, ISSN: 00295981. DOI: 10.1002/nme.5825.
- [5] I. Griбанov, R. Taylor, and R. Sarracino, “The gradient and the hessian of the distance between point and triangle in 3d,” *Algorithms*, vol. 11, no. 7, p. 104, Jul. 2018, ISSN: 1999-4893. DOI: 10.3390/a11070104.

A | Appendix: Reference Implementation for Calculating the Gradient and the Hessian of the Distance between a Point and a Triangle in 3D

```
// point-triangle derivatives; returns squared distance
// input: x[12] is the array of coordinates of 4 points:
// p0, p1, p2, p3
// p0 is the point from which the distance is computed,
// p1,p2,p3 form the triangle
// the function returns the squared distance
// fd is the output array of first derivatives of the
// squared distance
// sd is the output array of the second derivatives of
// the squared distance
// zeta2 and zeta3 are barycentric coordinates of the
// closest point on the triangle
// zeta1 can be obtained as 1-(zeta2+zeta3)
double pt(double (&x)[12], double (&fd)[12],
          double (&sd)[12][12],
          double &zeta2, double &zeta3);

// POINT-LINE
// dot product of the form (p1-p0)(p2-p1)
double sp_dot3(
double (&x)[9], // input: coords of p0,p1,p2
// must be cleared before calling this function
double (&fd)[9], // output: first derivatives
// must be cleared
double (&sd)[9][9]) // output: second derivatives;
{
double x0 = x[0];
```

```

double y0 = x[1];
double z0 = x[2];

double x1 = x[3];
double y1 = x[4];
double z1 = x[5];

double x2 = x[6];
double y2 = x[7];
double z2 = x[8];

fd[0] = x1 - x2;
fd[1] = y1 - y2;
fd[2] = z1 - z2;
fd[3] = x0 - 2 * x1 + x2;
fd[4] = y0 - 2 * y1 + y2;
fd[5] = z0 - 2 * z1 + z2;
fd[6] = x1 - x0;
fd[7] = y1 - y0;
fd[8] = z1 - z0;

// second derivs
for (int k = 0; k < 3; k++) {
    sd[k + 3][k + 3] = -2;
    sd[k][k + 6] = sd[k + 6][k] = -1;
    sd[k][k + 3] = sd[k + 3][k] =
        sd[k + 6][k + 3] = sd[k + 3][k + 6] = 1;
}
return (x1 - x0) * (x2 - x1) + (y1 - y0) * (y2 - y1) +
    (z1 - z0) * (z2 - z1);
}

// calculate the first and the second derivatives of f^2,
// given that f' and f'' are known
double function_squared(
double f,           // input: value at point
double (&fd)[9],    // input: first derivatives
double (&sd)[9][9], // input: second derivatives
double (&fdOut)[9], // output: first derivatives
double (&sdOut)[9][9]) // output: second derivatives
{
    for (int i = 0; i < 9; i++) {
        fdOut[i] = 2 * f * fd[i];
    }
}

```



```

        for (int j = 0; j < 9; j++)
            sdOut[i][j] = 2 * (fd[i] * fd[j] +
                               f * sd[i][j]);
    }
    return f * f;
}

double sp_dot3_squared(
double (&x)[9], // input: coords
double (&fd)[9], // output: first derivatives
double (&sd)[9][9]) // output: second derivatives
{
    double sp_dot3_fd[9] = { };
    double sp_dot3_sd[9][9] = { };
    double sp_dot3_value = sp_dot3(x, sp_dot3_fd,
                                     sp_dot3_sd);
    double result = function_squared(sp_dot3_value,
                                     sp_dot3_fd, sp_dot3_sd, fd, sd);
    return result;
}

// value, 1st and 2nd derivatives of the squared
// distance between points selected by idx1 and idx2
double vertex_vertex_distance_and_derivs(
int idx1,
int idx2,
double (&x)[9], // input: coords
double (&sdd)[9], // output: first derivatives
double (&sdd2)[9][9]) // output: second derivatives
{
    int ix0 = idx1 * 3 + 0;
    int iy0 = idx1 * 3 + 1;
    int iz0 = idx1 * 3 + 2;

    int ix1 = idx2 * 3 + 0;
    int iy1 = idx2 * 3 + 1;
    int iz1 = idx2 * 3 + 2;

    double x0 = x[ix0];
    double y0 = x[iy0];
    double z0 = x[iz0];

    double x1 = x[ix1];

```

```

double y1 = x[iy1];
double z1 = x[iz1];

sdd[ix0] = 2 * (x0 - x1);
sdd[iy0] = 2 * (y0 - y1);
sdd[iz0] = 2 * (z0 - z1);

sdd[ix1] = -sdd[ix0];
sdd[iy1] = -sdd[iy0];
sdd[iz1] = -sdd[iz0];

sdd2[ix0][ix0] = sdd2[iy0][iy0] = sdd2[iz0][iz0] = 2;
sdd2[ix1][ix1] = sdd2[iy1][iy1] = sdd2[iz1][iz1] = 2;
sdd2[ix0][ix1] = sdd2[iy0][iy1] = sdd2[iz0][iz1] = -2;
sdd2[ix1][ix0] = sdd2[iy1][iy0] = sdd2[iz1][iz0] = -2;

return (x0 - x1) * (x0 - x1) + (y0 - y1) * (y0 - y1) +
       (z0 - z1) * (z0 - z1);
}

// value, 1st and 2nd derivatives of the squared
// distance between points selected by idx1 and idx2
double vertex_vertex_distance_and_derivs_12(
int idx1,
int idx2,
double (&x)[12], // input: coords
double (&fd)[12], // output: first derivatives
double (&sd)[12][12]) // output: second derivatives
{
    int ix0 = idx1 * 3 + 0;
    int iy0 = idx1 * 3 + 1;
    int iz0 = idx1 * 3 + 2;

    int ix1 = idx2 * 3 + 0;
    int iy1 = idx2 * 3 + 1;
    int iz1 = idx2 * 3 + 2;

    double x0 = x[ix0];
    double y0 = x[iy0];
    double z0 = x[iz0];

    double x1 = x[ix1];
    double y1 = x[iy1];

```

```

double z1 = x[iz1];

fd[ix0] = 2 * (x0 - x1);
fd[iy0] = 2 * (y0 - y1);
fd[iz0] = 2 * (z0 - z1);

fd[ix1] = -fd[ix0];
fd[iy1] = -fd[iy0];
fd[iz1] = -fd[iz0];

sd[ix0][ix0] = sd[iy0][iy0] = sd[iz0][iz0] = 2;
sd[ix1][ix1] = sd[iy1][iy1] = sd[iz1][iz1] = 2;
sd[ix0][ix1] = sd[iy0][iy1] = sd[iz0][iz1] = -2;
sd[ix1][ix0] = sd[iy1][iy0] = sd[iz1][iz0] = -2;

return (x0 - x1) * (x0 - x1) + (y0 - y1) * (y0 - y1) +
       (z0 - z1) * (z0 - z1);
}

double vertex_edge_distance_and_derivs(
double (&x)[9], // input coords; p0, line:(p1, p2)
double (&sdd)[9], // output: first derivatives
double (&sdd2)[9][9]) // output: second derivatives
{
double x0 = x[0];
double x1 = x[1];
double x2 = x[2];
double x3 = x[3];
double x4 = x[4];
double x5 = x[5];
double x6 = x[6];
double x7 = x[7];
double x8 = x[8];

double edge_length_sq = (-x3 + x6) * (-x3 + x6) +
    (-x4 + x7) * (-x4 + x7)
    + (-x5 + x8) * (-x5 + x8);
double t = -((-x0 + x3) * (-x3 + x6) +
    (-x1 + x4) * (-x4 + x7)
    + (-x2 + x5) * (-x5 + x8)) / edge_length_sq;

double u = 1 - t;
double sqrDist = (x0 - u * x3 - t * x6) *

```

```

        (x0 - u * x3 - t * x6)
        + (x1 - u * x4 - t * x7) *
        (x1 - u * x4 - t * x7)
        + (x2 - u * x5 - t * x8) *
        (x2 - u * x5 - t * x8);

double g;
double g_fd[9] = { };
double g_sd[9][9] = { };

// |(x1-x0)(x2-x1)|^2 and its derivatives
g = sp_dot3_squared(x, g_fd, g_sd);

// f1 = |x1-x0|^2
double f1fd[9] = { };
double f1sd[9][9] = { };

// f2 = |x2-x1|^2
double f2;
double f2fd[9] = { };
double f2sd[9][9] = { };
f2 = vertex_vertex_distance_and_derivs(
    2, 1, x, f2fd, f2sd);

// combine together
double f2sq = f2 * f2;
double f2cube = f2sq * f2;
for (int i = 0; i < 9; i++) {
    sdd[i] = f1fd[i] + (g * f2fd[i] - f2 * g_fd[i])
        / f2sq;

    for (int j = 0; j < 9; j++) {
        double term1 = -2 * g * f2fd[i] * f2fd[j]
            / f2cube;
        double term2 = (g_fd[i] * f2fd[j] +
            g_fd[j] * f2fd[i]) / f2sq;
        double term3 = f1sd[i][j];
        double term4 = g * f2sd[i][j] / f2sq;
        double term5 = -g_sd[i][j] / f2;
        sdd2[i][j] = term1 + term2 + term3 +
            term4 + term5;
    }
}

```

```

    return sqrDist;
}

// version for arrays with 12 elements
double vertex_edge_distance_and_derivs_12(
double (&x)[12], // input coords; p0, line: (p1, p2)
int idx1,        // input: index for p1
int idx2,        // input: index for p2
double (&fd)[12], // output: first derivatives
double (&sd)[12][12]) // output: second derivatives
{
    double _x[9];
    _x[0] = x[0];
    _x[1] = x[1];
    _x[2] = x[2];

    idx1 *= 3;
    idx2 *= 3;

    double p01 = (x[0] - x[0 + idx1]) * (x[0] -
        x[0 + idx1])
        + (x[1] - x[1 + idx1]) * (x[1] - x[1 + idx1])
        + (x[2] - x[2 + idx1]) * (x[2] - x[2 + idx1]);
    double p02 = (x[0] - x[0 + idx2]) *
        (x[0] - x[0 + idx2])
        + (x[1] - x[1 + idx2]) * (x[1] - x[1 + idx2])
        + (x[2] - x[2 + idx2]) * (x[2] - x[2 + idx2]);

    if (p01 > p02) {
        // swap indices
        int tmp_idx = idx1;
        idx1 = idx2;
        idx2 = tmp_idx;
    }

    _x[3] = x[0 + idx1];
    _x[4] = x[1 + idx1];
    _x[5] = x[2 + idx1];

    _x[6] = x[0 + idx2];
    _x[7] = x[1 + idx2];
    _x[8] = x[2 + idx2];

```

```

double _fd[9] = { };
double _sd[9][9] = { };

double result = vertex_edge_distance_and_derivs(
    _x, _fd, _sd);

// distribute _fd and _sd

for (int i = 0; i < 3; i++) {
    fd[i] = _fd[i];
    fd[idx1 + i] = _fd[3 + i];
    fd[idx2 + i] = _fd[6 + i];

    for (int j = 0; j < 3; j++) {
        sd[i][j] = _sd[i][j];
        sd[i + idx1][j] = _sd[3 + i][j];
        sd[i][j + idx1] = _sd[i][3 + j];
        sd[i + idx1][j + idx1] = _sd[i + 3][j + 3];
        sd[i + idx1][j + idx2] = _sd[i + 3][j + 6];
        sd[i + idx2][j + idx1] = _sd[i + 6][j + 3];
        sd[i][j + idx2] = _sd[i][j + 6];
        sd[i + idx2][j] = _sd[i + 6][j];
        sd[i + idx2][j + idx2] = _sd[i + 6][j + 6];
    }
}

return result;
}

// POINT-PLANE (intended to use in the interior of the triangle)
// second derivatives of a
double a2[12][12] = {
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 2, 0, 0, -2, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 2, 0, 0, -2, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 2, 0, 0, -2, 0, 0, 0 },
    { 0, 0, 0, -2, 0, 0, 2, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, -2, 0, 0, 2, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, -2, 0, 0, 2, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
};

```

```

    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 } };

// second derivatives of b
double b2[12][12] = {
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 2, 0, 0, -1, 0, 0, -1, 0, 0 },
    { 0, 0, 0, 0, 2, 0, 0, -1, 0, 0, -1, 0 },
    { 0, 0, 0, 0, 0, 2, 0, 0, -1, 0, 0, -1 },
    { 0, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0, 0 },
    { 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0 },
    { 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 1 },
    { 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0 } };

// second derivatives of c
double c2[12][12] = {
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 2, 0, 0, 0, 0, 0, -2, 0, 0 },
    { 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, -2, 0 },
    { 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, -2 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, -2, 0, 0, 0, 0, 0, 2, 0, 0 },
    { 0, 0, 0, 0, -2, 0, 0, 0, 0, 0, 2, 0 },
    { 0, 0, 0, 0, 0, -2, 0, 0, 0, 0, 0, 2 } };

// second derivatives of d
double d2[12][12] = {
    { 0, 0, 0, 1, 0, 0, -1, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 1, 0, 0, -1, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 1, 0, 0, -1, 0, 0, 0 },
    { 1, 0, 0, -2, 0, 0, 1, 0, 0, 0, 0, 0 },
    { 0, 1, 0, 0, -2, 0, 0, 1, 0, 0, 0, 0 },
    { 0, 0, 1, 0, 0, -2, 0, 0, 1, 0, 0, 0 },
    { -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },

```

```

    { 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 } };

// second derivatives of e
double e2[12][12] = {
    { 0, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0, 0 },
    { 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0 },
    { 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, -1 },
    { 1, 0, 0, -2, 0, 0, 0, 0, 0, 1, 0, 0 },
    { 0, 1, 0, 0, -2, 0, 0, 0, 0, 0, 1, 0 },
    { 0, 0, 1, 0, 0, -2, 0, 0, 0, 0, 0, 1 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0 } };

// second derivatives of f
double f2[12][12] = {
    { 2, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 2, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 2, 0, 0, -2, 0, 0, 0, 0, 0, 0 },
    { -2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, -2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, -2, 0, 0, 2, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 } };

// Kronecker delta
double xd(int idx1, int idx2) {
    return idx1 == idx2 ? 1. : 0;
}

// input: x[12] array of p0,p1,p2,p3
// output: first derivatives of squared distance fd[12]
// second derivatives of squared distance sd[12][12]

```



```

// return value is the squared distance
// note: don't call this function directly
double point_plane_distance(double (&x)[12], double (&fd)[12],
    double (&sd)[12][12]) {
    double x0 = x[0];
    double x1 = x[1];
    double x2 = x[2];
    double x3 = x[3];
    double x4 = x[4];
    double x5 = x[5];
    double x6 = x[6];
    double x7 = x[7];
    double x8 = x[8];
    double x9 = x[9];
    double x10 = x[10];
    double x11 = x[11];

    double a = (-x3 + x6) * (-x3 + x6) +
        (-x4 + x7) * (-x4 + x7)
        + (-x5 + x8) * (-x5 + x8);
    double b = (x10 - x4) * (-x4 + x7) +
        (x11 - x5) * (-x5 + x8)
        + (-x3 + x6) * (-x3 + x9);
    double c = (x10 - x4) * (x10 - x4) +
        (x11 - x5) * (x11 - x5)
        + (-x3 + x9) * (-x3 + x9);
    double d = (-x0 + x3) * (-x3 + x6) +
        (-x1 + x4) * (-x4 + x7)
        + (-x2 + x5) * (-x5 + x8);
    double e = (x10 - x4) * (-x1 + x4) +
        (x11 - x5) * (-x2 + x5)
        + (-x0 + x3) * (-x3 + x9);

    double det = a * c - b * b;
    double detsq = det * det;
    double detcube = detsq * det;

    double s = b * e - c * d;
    double t = b * d - a * e;

    double invDet = 1. / det;
    s *= invDet;
    t *= invDet;
}

```

```

double u = 1 - (s + t);

double sqrDistance = (-x0 + x6 * s + x9 * t + x3 * u)
    * (-x0 + x6 * s + x9 * t + x3 * u)
    + (-x1 + x7 * s + x10 * t + x4 * u)
        * (-x1 + x7 * s + x10 * t + x4 * u)
    + (-x2 + x8 * s + x11 * t + x5 * u)
        * (-x2 + x8 * s + x11 * t + x5 * u);

// u = zeta1; s = zeta2; t = zeta3;
double s2[12][12], t2[12][12], det2[12][12];
double u1[12], u2[12][12];

// derivatives of s and t
// first derivatives of the above quantities
double a1[12] = { 0, 0, 0, -2 * (-x3 + x6),
    -2 * (-x4 + x7), -2
    * (-x5 + x8), 2 * (-x3 + x6), 2 * (-x4 + x7),
    2 * (-x5 + x8), 0, 0, 0 };
double b1[12] = { 0, 0, 0, 2 * x3 - x6 - x9,
    -x10 + 2 * x4 - x7, -x11
    + 2 * x5 - x8, -x3 + x9, x10 - x4, x11 - x5,
    -x3 + x6, -x4 + x7, -x5 + x8 };
double c1[12] = { 0, 0, 0, -2 * (-x3 + x9),
    -2 * (x10 - x4), -2
    * (x11 - x5), 0, 0, 0, 2 * (-x3 + x9),
    2 * (x10 - x4), 2
    * (x11 - x5) };
double d1[12] = { x3 - x6, x4 - x7, x5 - x8,
    x0 - 2 * x3 + x6, x1 - 2 * x4
    + x7, x2 - 2 * x5 + x8, -x0 + x3, -x1 + x4,
    -x2 + x5, 0, 0, 0 };
double e1[12] = { x3 - x9, -x10 + x4, -x11 + x5,
    x0 - 2 * x3 + x9, x1 + x10
    - 2 * x4, x11 + x2 - 2 * x5, 0, 0, 0,
    -x0 + x3, -x1 + x4, -x2 + x5 };
double s1[12], t1[12], det1[12];

// first derivatives
for (int i = 0; i < 12; i++) {
    det1[i] = c * a1[i] + a * c1[i] - 2 * b * b1[i];
    s1[i] = ((c * d - b * e) * det1[i]) / detsq
        + ((e * b1[i] + b * e1[i]) -

```

```

        (d * c1[i] + c * d1[i])) / det;
t1[i] = ((a * e - b * d) * det1[i]) / detsq
        + ((d * b1[i] + b * d1[i]) -
          (a * e1[i] + e * a1[i])) / det;
u1[i] = -(s1[i] + t1[i]);

// expression is machine-generated
fd[i] = -2 * (x0 - x6 * s - x9 * t - x3 * u)
* (x6 * s1[i] + x9 * t1[i] + x3 * u1[i] - xd(0, i)
  + u * xd(3, i) + s * xd(6, i) + t * xd(9, i))
- 2 * (x1 - x7 * s - x10 * t - x4 * u)
* (x7 * s1[i] + x10 * t1[i] + x4 * u1[i] - xd(1, i)
  + u * xd(4, i) + s * xd(7, i) + t * xd(10, i))
- 2 * (x2 - x8 * s - x11 * t - x5 * u)
* (x8 * s1[i] + x11 * t1[i] + x5 * u1[i] - xd(2, i)
  + u * xd(5, i) + s * xd(8, i) + t * xd(11, i));
}

for (int i = 0; i < 12; i++)
  for (int j = 0; j < 12; j++) {
    det2[i][j] =
      -2 * b1[i] * b1[j] + a1[j] * c1[i] +
      a1[i] * c1[j]
      + c * a2[i][j] - 2 * b * b2[i][j] +
      a * c2[i][j];

    s2[i][j] = +(-(c1[j] * d1[i]) -
      c1[i] * d1[j] + b1[j] * e1[i]
      + b1[i] * e1[j] + e * b2[i][j] -
      d * c2[i][j] - c * d2[i][j]
      + b * e2[i][j]) / det
      - ((det1[j]
        * (e * b1[i] - d * c1[i] - c * d1[i] +
          b * e1[i])) + (det1[i]
          * (e * b1[j] - d * c1[j] - c * d1[j]
            + b * e1[j]))
        + ((-(c * d) + b * e) * det2[i][j])) / detsq
      + (2 * (-(c * d) + b * e) *
        det1[i] * det1[j]) / detcube;

    t2[i][j] =
      +(b1[j] * d1[i] + b1[i] * d1[j] -
        a1[j] * e1[i]

```

```

- a1[i] * e1[j] - e * a2[i][j] + d * b2[i][j]
+ b * d2[i][j] - a * e2[i][j]) / det
- ((det1[j] * (-(e * a1[i]) +
d * b1[i] + b * d1[i]
- a * e1[i])) + (det1[i]
* (-(e * a1[j]) + d * b1[j]
+ b * d1[j] - a * e1[j]))
+ ((b * d - a * e) * det2[i][j])) / detsq
+ (2 * (b * d - a * e) * det1[i] * det1[j])
/ detcube;

```

```

u2[i][j] = -(s2[i][j] + t2[i][j]);

```

```

// this expression was machine-generated
sd[i][j] =
2 * ((x6 * s1[i] + x9 * t1[i] + x3 * u1[i] -
xd(0, i)
+ u * xd(3, i) + s * xd(6, i) + t * xd(9, i))
* (x6 * s1[j] + x9 * t1[j] + x3 * u1[j]
- xd(0, j) + u * xd(3, j)
+ s * xd(6, j) + t * xd(9, j))
- (x0 - x6 * s - x9 * t - x3 * u)
* (x6 * s2[i][j] + x9 * t2[i][j]
+ x3 * u2[i][j]
+ u1[j] * xd(3, i)
+ u1[i] * xd(3, j)
+ s1[j] * xd(6, i)
+ s1[i] * xd(6, j)
+ t1[j] * xd(9, i)
+ t1[i] * xd(9, j))
+ (x7 * s1[i] + x10 * t1[i] + x4 * u1[i]
- xd(1, i) + u * xd(4, i)
+ s * xd(7, i) + t * xd(10, i))
* (x7 * s1[j] + x10 * t1[j]
+ x4 * u1[j] - xd(1, j)
+ u * xd(4, j)
+ s * xd(7, j)
+ t * xd(10, j))
- (x1 - x7 * s - x10 * t - x4 * u)
* (x7 * s2[i][j] + x10 * t2[i][j]
+ x4 * u2[i][j]
+ u1[j] * xd(4, i)
+ u1[i] * xd(4, j)

```

```

+ s1[j] * xd(7, i)
+ s1[i] * xd(7, j)
+ t1[j] * xd(10, i)
+ t1[i] * xd(10, j))
+ (x8 * s1[i] + x11 * t1[i] + x5 * u1[i]
- xd(2, i) + u * xd(5, i)
+ s * xd(8, i) + t * xd(11, i))
* (x8 * s1[j] + x11 * t1[j]
+ x5 * u1[j] - xd(2, j)
+ u * xd(5, j)
+ s * xd(8, j)
+ t * xd(11, j))
- (x2 - x8 * s - x11 * t - x5 * u)
* (x8 * s2[i][j] + x11 * t2[i][j]
+ x5 * u2[i][j]
+ u1[j] * xd(5, i)
+ u1[i] * xd(5, j)
+ s1[j] * xd(8, i)
+ s1[i] * xd(8, j)
+ t1[j] * xd(11, i)
+ t1[i] * xd(11, j))));
}

return sqrDistance;
}

// POINT-TRIANGE (ALL CASES COMBINED)
double pt(double (&x)[12], double (&fd)[12],
double (&sd)[12][12],
double &zeta2, double &zeta3) {
double x0 = x[0];
double x1 = x[1];
double x2 = x[2];
double x3 = x[3];
double x4 = x[4];
double x5 = x[5];
double x6 = x[6];
double x7 = x[7];
double x8 = x[8];
double x9 = x[9];
double x10 = x[10];
double x11 = x[11];

```

```

double a = (-x3 + x6) * (-x3 + x6) +
    (-x4 + x7) * (-x4 + x7)
    + (-x5 + x8) * (-x5 + x8);
double b = (x10 - x4) * (-x4 + x7) +
    (x11 - x5) * (-x5 + x8)
    + (-x3 + x6) * (-x3 + x9);
double c = (x10 - x4) * (x10 - x4) +
    (x11 - x5) * (x11 - x5)
    + (-x3 + x9) * (-x3 + x9);
double d = (-x0 + x3) * (-x3 + x6) +
    (-x1 + x4) * (-x4 + x7)
    + (-x2 + x5) * (-x5 + x8);
double e = (x10 - x4) * (-x1 + x4) +
    (x11 - x5) * (-x2 + x5)
    + (-x0 + x3) * (-x3 + x9);

double det = a * c - b * b;
double s = b * e - c * d;
double t = b * d - a * e;
double sqrDistance;

if (s + t <= det) {
    if (s < 0) {
        if (t < 0) // region 4
        {
            if (d < 0) {
                t = 0;
                if (-d >= a) {
                    s = 1;
                    sqrDistance =
                        vertex_vertex_distance_and_derivs_12(
                            0, 2, x, fd, sd);
                } else {
                    s = -d / a;
                    sqrDistance =
                        vertex_edge_distance_and_derivs_12(
                            x, 2, 1, fd, sd);
                }
            } else {
                s = 0;
                if (e >= 0) {
                    t = 0;
                    sqrDistance =

```

```

        vertex_vertex_distance_and_derivs_12(
            0, 1, x, fd, sd);
    } else if (-e >= c) {
        t = 1;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 3, x, fd, sd);
    } else {
        t = -e / c;
        sqrDistance =
            vertex_edge_distance_and_derivs_12(
                x, 3, 1, fd, sd);
    }
}
} else // region 3
{
    s = 0;
    if (e >= 0) {
        t = 0;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 1, x, fd, sd);
    } else if (-e >= c) {
        t = 1;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 3, x, fd, sd);
    } else {
        t = -e / c;
        sqrDistance =
            vertex_edge_distance_and_derivs_12(
                x, 3, 1, fd, sd);
    }
}
} else if (t < 0) // region 5
{
    t = 0;
    if (d >= 0) {
        s = 0;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 1, x, fd, sd);
    } else if (-d >= a) {

```

```

        s = 1;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 2, x, fd, sd);
    } else {
        s = -d / a;
        sqrDistance =
            vertex_edge_distance_and_derivs_12(
                x, 1, 2, fd, sd);
    }
} else // region 0
{
    double invDet = (1) / det;
    s *= invDet;
    t *= invDet;
    sqrDistance = point_plane_distance(x, fd, sd);
}
} else {
    double tmp0, tmp1, numer, denom;

    if (s < 0) // region 2
    {
        tmp0 = b + d;
        tmp1 = c + e;
        if (tmp1 > tmp0) {
            numer = tmp1 - tmp0;
            denom = a - 2 * b + c;
            if (numer >= denom) {
                s = 1;
                t = 0;
                sqrDistance =
                    vertex_vertex_distance_and_derivs_12(
                        0, 2, x, fd, sd);
            } else {
                s = numer / denom;
                t = 1 - s;
                sqrDistance =
                    vertex_edge_distance_and_derivs_12(
                        x, 2, 3, fd, sd);
            }
        } else {
            s = 0;
            if (tmp1 <= 0) {

```



```

        t = 1;
        sqrDistance =
        vertex_vertex_distance_and_derivs_12(
        0, 3, x, fd, sd);
    } else if (e >= 0) {
        t = 0;
        sqrDistance =
        vertex_vertex_distance_and_derivs_12(
        0, 1, x, fd, sd);
    } else {
        t = -e / c;
        sqrDistance =
        vertex_edge_distance_and_derivs_12(
        x, 1, 3, fd, sd);
    }
}
} else if (t < 0) // region 6
{
    tmp0 = b + e;
    tmp1 = a + d;
    if (tmp1 > tmp0) {
        numer = tmp1 - tmp0;
        denom = a - 2 * b + c;
        if (numer >= denom) {
            t = 1;
            s = 0;
            sqrDistance =
            vertex_vertex_distance_and_derivs_12(
            0, 3, x, fd, sd);
        } else {
            t = numer / denom;
            s = 1 - t;
            sqrDistance =
            vertex_edge_distance_and_derivs_12(
            x, 2, 3, fd, sd);
        }
    } else {
        t = 0;
        if (tmp1 <= 0) {
            s = 1;
            sqrDistance =
            vertex_vertex_distance_and_derivs_12(
            0, 2, x, fd, sd);

```

```

    } else if (d >= 0) {
        s = 0;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 1, x, fd, sd);
    } else {
        s = -d / a;
        sqrDistance =
            vertex_edge_distance_and_derivs_12(
                x, 1, 2, fd, sd);
    }
}
} else // region 1
{
    numer = c + e - b - d;
    if (numer <= 0) {
        s = 0;
        t = 1;
        sqrDistance =
            vertex_vertex_distance_and_derivs_12(
                0, 3, x, fd, sd);
    } else {
        denom = a - 2 * b + c;
        if (numer >= denom) {
            s = 1;
            t = 0;
            sqrDistance =
                vertex_vertex_distance_and_derivs_12(
                    0, 2, x, fd, sd);
        } else {
            s = numer / denom;
            t = 1 - s;
            sqrDistance =
                vertex_edge_distance_and_derivs_12(
                    x, 2, 3, fd, sd);
        }
    }
}
}
}
zeta2 = s;
zeta3 = t;
return sqrDistance;
}

```