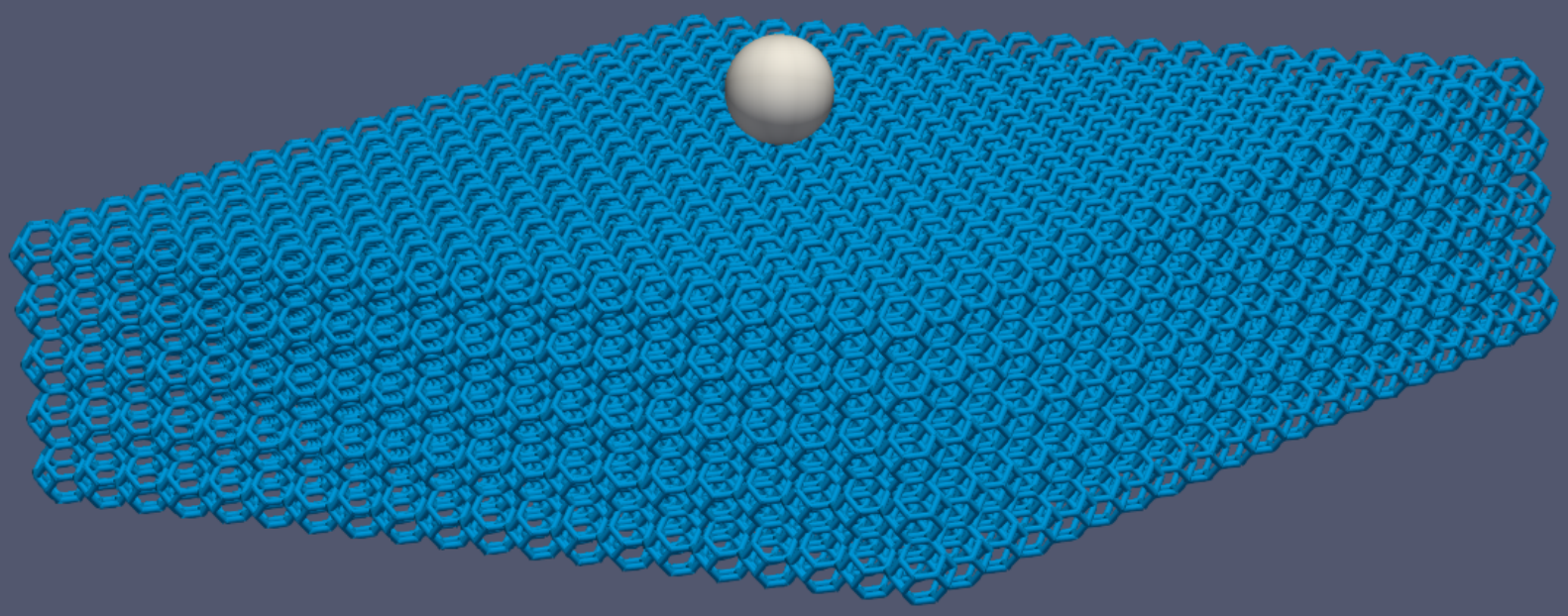
Computational modelling of impact in architected materials

A finite element framework for contact and fracture in beam lattices

Jimmy Chen



TU Delft

Computational modelling of impact in architected materials

A finite element framework for contact and fracture in beam lattices

by

Jimmy Chen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday 14 October, 2026 at 13:00.

Student number:	5580234	
Project duration:	6 January, 2026 – 30 September, 2026	
Thesis committee:	Dr. B. Giovanardi,	TU Delft, supervisor
	Dr. D. Peeters,	TU Delft, chair
	Dr. S. Turteltaub,	TU Delft, examiner

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Acknowledgements

I am extremely grateful to my thesis supervisor, Dr Bianca Giovanardi, for her guidance, patience, flexibility and insight. She was always readily available for help, and her experience and insights have proved invaluable on multiple occasions over the course of this thesis. Her enthusiasm, optimism and wholehearted support often had alleviating effects when I was not necessarily in a positive mood. I cannot stress enough how crucial she has been for this endeavour.

I would also like to extend my sincerest thanks to my family. I am not a person who counts my blessings, much less vocalises them, but I would like to make clear that although I never mention it, their presence alone brings a level of reassurance impossible to express in the human tongue. Lastly, I would like to mention my friends. Some of whom I have known since secondary school; others I have met since starting university. To them, thank you for giving my everyday life colour through light-hearted moments, immortalised inside jokes and (not always) intellectual discussions over various topics.

Jimmy Chen
Delft, September 2026

Executive Summary

Metamaterials have recently risen to become a highly intriguing class of materials. They boast unique material properties and are known to be extremely tailorable. Some examples of their unique material properties include negative refractive index and negative Poisson's ratio [1]. However, another exceptional property of metamaterials is their energy absorption efficiency, enabled by a combination of a low relative volume density, a tendency to stiffen under compression and tailorable mechanical properties [2] [3]. The energy absorption performance of metamaterials makes them prime candidates for advanced impact protection. High-performance fields like the aerospace industry place high emphasis on impact protection, both for aircraft and spacecraft. Space structures must contend with numerous particles travelling at extremely high velocities, and the amount of debris will only grow [4]. Meanwhile, the aviation industry loses upwards of billions of euros annually due to impact damage [5]. This highlights the relevancy of metamaterials as impact protection, but with this also comes a need for computational models of impact in metamaterials. Numerical models play a crucial role throughout the design process of an impact-resistant material, allowing designers to make better informed choices. As such, accurate simulations of impact performance of metamaterials can bring about significant cost and time savings.

When specifically looking at truss metamaterials, such a computational model must describe the dynamics of each strut, the failure behaviour of the strut material, and possible contact between struts. However, there is no one definitive state-of-the-art model for modelling impact in truss-lattice materials. To describe the dynamics of the struts, the literature is divided between continuum elements and beam elements. This choice of elements has great implications for the choice of contact model since models for beam contact are implemented differently from contact models for solid elements. As for modelling the failure of the lattice members, the literature usually employs damage evolution models to degrade and eventually delete damaged elements. However, deleting elements comes with the downside that contact interactions between the fragments and the intact struts cannot be captured. From the above, the following research gap in modelling impact in truss metamaterials is identified: a beam-based model that incorporates both beam contact and beam fracture.

To this end, this thesis develops a unified contact and fracture finite element framework for simulating dynamic impact with beam-based truss metamaterials. The primary objectives of this study are developing a unified beam-contact-fracture framework, conducting simple numerical experiments to test its functionality, and applying the framework to a large-scale impact problem involving a truss metamaterial to evaluate the effect of self-contact on the energy absorption. The aim of this research is to develop high-fidelity dynamic impact simulations of beam lattices capturing both strut self-contact and fracture behaviour.

Central to the aim of developing the unified contact and fracture computational framework for simulating truss-metamaterial impact problems, is synthesising a beam-contact-fracture finite element model to describe the behaviour of individual lattice members. Then, to extend the framework to large-scale impact simulations of entire lattice structures, this thesis addresses the modelling approach for the lattice joints, the contact formulation between the impactor and the lattice, and the parallelisation of the impact problem. In addition, the quantification of the energy components is discussed. Decomposing the total energy into separate energy contributions such as internal, interface and kinetic energy offers a method of assessing the performance of the computational framework.

To test the functionality of the unified beam-contact-fracture framework, it was employed in three numerical experiments. Each numerical experiment featured either just one unit cell, or in the case of the third experiment, two unit cells. The aim of the numerical experiments was to assess the performance of the computational framework in stages. The first test case was the gradual compression of a unit cell to evaluate the contact and fracture behaviour of the struts under quasi-static loading. The second test case involved dynamic loading by applying an initial velocity front to a single unit cell. For this

case, the fracture behaviour and energy balance under these loading conditions were analysed. The third case modelled the collision of two unit cells to test everything at once: contact and fracture due to collision along with energy balance.

After the numerical experiments, the unified beam-contact-fracture framework was deployed to a large-scale truss-metamaterial impact problem. The lattice in this experiment comprised 500 unit cells, far larger than the abovementioned numerical experiments. The larger problem allows for more complexities than is possible with the collision of two unit cells, for instance. To realise simulations of this size, the problem was run on a high-performance computing cluster to take advantage of parallel computing. Six test cases in total were simulated. Three accounted for self-contact and three did not. Four of the test cases used lattices comprised of bitruncated octahedron unit cells while the other two used lattices made from octet unit cells. The bitruncated octahedron test cases were simulated with initial impactor velocities $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s while the octet test cases were only simulated with an initial impactor velocity $v_0 = 350.0$ m/s due to time constraints.

Since the aim of this research was to develop high-fidelity dynamic impact simulations of beam lattices capturing both strut self-contact and fracture behaviour, the results of the test cases with self-contact were compared with the simulation runs where no self-contact between the struts was modelled. The results came in the form of impactor trajectories and the energy responses. These were then plotted to illustrate the trajectory of the impactor for each simulation run, the energy components of the system and the energy absorption of each test case. Through these plots, comparisons between the contact-on and contact-off cases were possible.

When the mass-normalised absorbed energy of each contact-on case was compared with its associated contact-off case, it turned out that modelling self-contact reduced the energy absorption performance of the lattices. For the bitruncated octahedron lattices, for $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s, the loss in mass-normalised absorbed energy of the cases with self-contact compared with the cases without was approximately 12%. Meanwhile, the loss in mass-normalised absorbed energy of the octet lattice with self-contact compared with the one without for $v_0 = 350.0$ m/s was approximately 4.5%. This reduction in energy absorption performance was contrary to expectations since modelling self-contact should allow for capturing compaction behaviour of the lattice. However, because inter-partition contact cannot be resolved by the current implementation of the contact model, the simulations would never be able to fully capture compaction. As a result, implementing a fully parallel contact model that can resolve inter-partition contact is an important recommendation for future work.

This thesis closes out by highlighting and explaining important improvements for future work. These can build upon the work done in this study and enhance the functionality of the computational framework to make it more robust, more widely applicable, more accurate. Some examples of recommendations are devising a way of fully parallelising the contact model, implementing a line-to-line contact model, and modelling more impact phenomena such as plasticity, hydrodynamics or thermo-mechanics. In all, this thesis managed to fulfil the primary objectives, although it also revealed a key limitation of the developed computational framework, and through the recommendations, this thesis indicates potential avenues of improvement for high-fidelity impact simulations of beam lattices.

Contents

Acknowledgements i

Executive Summary ii

1 Introduction 1
- 1.1 Architected cellular materials . . . 1
- 1.2 Modelling impact in metamaterials . . . 2
- 1.3 Research approach . . . 3

2 Literature review 4
- 2.1 Architected cellular materials . . . 4
  - 2.1.1 Metamaterial classification . . . 4
  - 2.1.2 Crystal lattice metamaterials . . . 6
  - 2.1.3 Mechanical properties . . . 9
  - 2.1.4 Applications of mechanical metamaterials . . . 10
  - 2.1.5 Summary of the architected cellular materials review . . . 12
- 2.2 Impact experiments of truss-lattice metamaterials . . . 12
  - 2.2.1 Portela, Edwards, Veysset, Sun, Nelson, Kochmann and Greer . . . 12
  - 2.2.2 Butruille, Crone and Portela . . . 15
  - 2.2.3 Bieler and Weinberg . . . 20
  - 2.2.4 Summary of impact experiments of truss-lattice metamaterials review . . . 23
- 2.3 Impact physics . . . 24
  - 2.3.1 Impact regimes . . . 24
  - 2.3.2 Low velocity impact . . . 24
  - 2.3.3 High velocity impact . . . 26
  - 2.3.4 Energy absorption mechanisms . . . 27
  - 2.3.5 Summary of impact physics . . . 27
- 2.4 Computational modelling approaches for truss-metamaterial impact problems . . . 27
  - 2.4.1 Choice of elements . . . 28
  - 2.4.2 Modelling inter-element contact . . . 28
  - 2.4.3 Representing damage . . . 28
  - 2.4.4 Conclusion on the state-of-the-art of metamaterial impact modelling . . . 29
- 2.5 Overview of beam theories . . . 29
  - 2.5.1 Euler-Bernoulli theory . . . 30
  - 2.5.2 Timoshenko theory . . . 31
  - 2.5.3 Simo-Reissner theory . . . 31
  - 2.5.4 Geometrically nonlinear Kirchhoff rod theory . . . 36
  - 2.5.5 Torsion-free geometrically nonlinear Kirchhoff rod theory . . . 39
  - 2.5.6 Summary of the beam theories . . . 42
- 2.6 Contact mechanics . . . 42
  - 2.6.1 The general contact problem . . . 42
  - 2.6.2 Contact in hyperelastic materials . . . 44
  - 2.6.3 Enforcing contact constraints . . . 45
  - 2.6.4 Lagrange multiplier method . . . 46
  - 2.6.5 Penalty method . . . 48
  - 2.6.6 Formulating the beam-to-beam contact problem . . . 50
  - 2.6.7 Summary of contact mechanics review . . . 57
- 2.7 Fracture mechanics . . . 57
  - 2.7.1 Computational fracture mechanics . . . 57
  - 2.7.2 Cohesive zone model . . . 57

2.7.3 Modelling fracture in beams . . . 58
2.7.4 Summary of fracture mechanics review . . . 60

3 Methodology **62**
3.1 Synthesis of a unified contact and beam fracture framework . . . 62
3.1.1 Choice of beam model for truss architected materials . . . 62
3.1.2 The continuum formulation of the dynamic problem . . . 65
3.1.3 Introducing fracture behaviour into the beam mechanics . . . 67
3.1.4 Spatial and temporal discretisation . . . 70
3.1.5 Choice of beam-to-beam contact formulation . . . 72
3.1.6 Phase angle coordinate transformation . . . 73
3.1.7 Contact model . . . 75
3.1.8 Tying contact with the system dynamics . . . 79
3.1.9 Key assumptions . . . 80
3.2 Modelling a truss-lattice structure for impact problems . . . 80
3.2.1 The unit cell . . . 80
3.2.2 Modelling joint behaviour . . . 81
3.2.3 Resolving contact between a rigid impactor and the lattice . . . 82
3.2.4 Parallelisation . . . 84
3.2.5 Key assumptions . . . 86
3.3 Quantifying energy components . . . 86
3.3.1 Energy balance of a mechanical system . . . 86
3.3.2 Kinetic energy . . . 87
3.3.3 Internal energy . . . 88
3.3.4 External work . . . 88
3.3.5 Interface energy . . . 90
3.3.6 Assumptions and potential sources of error . . . 91
3.4 The SUMMIT framework . . . 91

4 Numerical experiments **94**
4.1 Case I: compression of a unit insideBCC . . . 94
4.1.1 Set-up and the insideBCC unit cell . . . 94
4.1.2 Results . . . 96
4.2 Case II: unit insideBCC subjected to an initial velocity front . . . 99
4.2.1 Results . . . 100
4.3 Case III: collision of two insideBCC unit cells . . . 105
4.3.1 Results . . . 107
4.3.2 Sensitivity study of R_{contact} . . . 109
4.3.3 Convergence study with Δt . . . 112
4.4 Case IIIb: collision of two insideBCC unit cells at higher velocity . . . 113
4.4.1 Results . . . 114
4.5 Comments on the test applications . . . 118

5 Results and discussion **119**
5.1 Simulation set-up . . . 119
5.2 Collision of a rigid impactor with a truss lattice . . . 122
5.2.1 Impact visualisations . . . 122
5.2.2 Trajectory plots . . . 128
5.2.3 Energy response . . . 132
5.2.4 Summary of results . . . 138

6 Conclusion **140**
6.1 Summary of findings . . . 140
6.2 Recommendations for future work . . . 141
6.2.1 Convergence study of the self-contact penalty parameter . . . 141
6.2.2 Alternative to nodal kinetic energy . . . 141
6.2.3 Direct nodal residual contribution of contact point-forces . . . 142
6.2.4 Inter-partition contact detection . . . 142

6.2.5 Alternatives to penalty contact . . . 142
6.2.6 Implementation of line-to-line contact . . . 143
6.2.7 Accounting for more impact phenomena . . . 144
6.2.8 Non-slender beam formulations . . . 145
6.2.9 Preventing boundary effects . . . 145
6.2.10 Expanding the scope of the impact tests . . . 145

7 Declaration of generative AI and AI tools **147**

References **152**

A Additional figures **153**
A.1 Additional figures for Subsection 4.3.2 . . . 153
A.2 Additional figures for Subsection 4.3.3 . . . 154

B Template license **156**

List of Figures

1.1 The evolution of the number of objects in all orbits, from [4]. The acronyms mean the following: UI means unidentified objects, RM is rocket mission related objects, RD means rocket debris, RF is rocket fragmentation debris, RB means rocket body, PM signifies payload related objects, PD is payload debris, PF means payload fragmentation debris, and PL is payload. 2

2.1 The different types of metamaterials and the associated area of application, from [18]. . 5
2.2 The different types of metamaterials according to Kumar et al. (2022) [19]. 5
2.3 A classification of cellular materials based on structure, from [6]. 6
2.4 The different structural configurations of lattice structures, from [26]. 7
2.5 The construction of a Wigner-Seitz cell, from [49]. 8
2.6 A unit cell and its possible node positions, from [6]. 8
2.7 The simple cubic cell (a) and the body-centred cubic cell (b), from [34]. 8
2.8 The face-centred cubic cell, made in GeoGebra. 8
2.9 The octet unit cell, from [6]. 9
2.10 The octahedron unit cell, made in GeoGebra. 9
2.11 The truncated octahedron unit cell, from [6]. 9
2.12 An Ashby plot which shows white regions unoccupied by commercial materials, and some cellular materials that can exist in the white regions, from [2]. 10
2.13 A schematic of the test set-up showing the LIPIT, the specimen and the camera, from [64]. 13
2.14 The impact responses and the resulting damage in the specimen with $\bar{\rho} = 23\%$. (a) shows impact and rebound while (b) illustrates particle capture. (c) is a sample after elastic impact. (d) is a post-impact micrograph of the sample from (a) which shows cratering, and (e) is a micrograph of the sample from (b) with the captured particle visible. The white scale bar equates to 20 μm, while the black scale bar is 5 μm. This figure can also be found in [64]. 14
2.15 The impact energies for the experiments. (a) shows the results for the specimen with $\bar{\rho} = 14\% \pm 3\%$, and (b) is for the specimen with $\bar{\rho} = 23\% \pm 4\%$. A linear fit of the inelastic energy W_i as a function of W_0 can be seen as the red dashed lines. [64] 14
2.16 A comparison of the mass-normalised inelastic energy W_i^* as a function of v_0 with other ballistic materials, from [64]. 15
2.17 The suspended microparticle impact framework described by Butruille et al. (2024). (A) shows the fabrication of the suspended metamaterial structure using photon lithography. The result of this can be seen in (B). (B i, ii and iii) are SEM micrographs the octet, monolithic and tetrakaidecahedron structures, respectively. The white scale bars indicate 100 μm and the yellow scale bars are 5 μm. (C) is a schematic of the impact experiments with the LIPIT method. This figure can also be found in [12]. 16
2.18 Micrographs of the impact experiments. (A) is for the octet structure, (B) is for the tetrakaidecahedron structure and (C) is for the monolithic sample. (D), (E) and (F) are post-impact SEM micrographs for the octet, tetrakaidecahedron and monolithic sample, respectively. [12] 16
2.19 The results of the microparticle impact tests. (A) shows the quantities that can be measured. (B) plots the final impact velocity to the initial impact velocity. (C) is a schematic of how the crater mass is calculated to normalise the dissipated energy. (D) plots the mass-normalised dissipated energy against the impact velocity. These results can also be found in [12]. 17

2.20 The decoupled compaction mechanism of microlattices. (A) is a schematic of the energy dissipation components during impact, showing W_f due to fracture, W_c due to compaction and W_{in} due to inertial effects. (B) and (C) show the simulated and experimental compaction behaviours of an octet and tetrakaidecahedron lattice, respectively. (D) and (E) depict the simulated and experimental stress-strain curves for both lattices. The scale bar equates to 20 μm. [12] . . . 18
2.21 The experiments and simulations conducted on quasi-static coupled compaction-fracture behaviour. (A, i) depict the CAD model and SEM images of the experiment for an octet and a tetrakaidecahedron lattice. The scale bar indicates 10 μm. (A, ii and iii) show the stress against the normalised displacement for the two lattices for varying pillar diameters. (B) are some capture frames of the corresponding simulations for the two lattices at normalised displacements 0.3, 0.45 and 0.6. (C, i) illustrates the deformation state of the octet lattice. (C, ii) depicts a simulation of a single octet and tetrakaidecahedron unit cell subjected to pure shear. (C, iii) show the stress-strain curves of the aforementioned simulations. (D) plots the normalised dissipated energy under the pillar-indentation loading. In the same plot, the normalised dissipated energy of the compaction tests are shown as dashed lines. [12] . . . 19
2.22 The diamond unit cell, from [6]. . . . 20
2.23 The rhombic dodecahedron unit cell, from [6]. . . . 21
2.24 The modified SHB set-up, from [6]. . . . 21
2.25 The cumulative impulse for the octet lattice for the three relative densities, from [6] . . . 22
2.26 The absorbed energy for each lattice of each relative density, from [6] . . . 23
2.27 The specific absorbed energy for each lattice at each relative density, from [6] . . . 23
2.28 Perforation mechanisms, from [66]. . . . 25
2.29 The deformation mode of a bending beam under the shear-rigid assumption from [71]. 30
2.30 The superposition of the Euler-Bernoulli beam and the shear deformation, resulting in a Timoshenko beam [71]. . . . 31
2.31 Kinematic quantities defining the initial and deformed configuration of the Simo-Reissner beam, from [69]. . . . 32
2.32 The kinematic quantities for the description a geometrically nonlinear Kirchhoff beam in its initial configuration, B_0, and current configuration, B, from [81]. . . . 36
2.33 Unilateral contact of an elastic solid, from [82]. . . . 42
2.34 Point mass supported by a spring, from [82]. . . . 46
2.35 Energy of the mass spring system, from [82]. . . . 46
2.36 The contact force represented by a Lagrange multiplier, from [82]. . . . 47
2.37 The point mass supported by a spring and a penalty spring due to the penalty term, from [82]. . . . 49
2.38 A general beam-contact problem, from [85]. . . . 50
2.39 Kinematic quantities for the point-to-point contact problem, from [17]. . . . 51
2.40 Some examples of contact problems with non-unique bilateral distance solutions, from [17]. 53
2.41 Kinematic quantities for the line-to-line contact problem, from [17]. . . . 54
2.42 Kinematic quantities for the discretised line-to-line contact problem, from [17]. . . . 55
2.43 The fracture modes of beams: (a) tension, (b) shear, (c) torsion and (d) bending, from [16]. 58
2.44 A linear traction-separation law for modelling beam fracture, from [16]. . . . 60

3.1 The discontinuous Galerkin discretisation of a straight and undeformed beam, from [16]. Internal nodes such as s_1^e and s_0^{e+1} are duplicated to capture discontinuities at element interfaces denoted by s_n with n the number of interfaces. . . . 67
3.2 A flowchart of the nearest neighbour search subroutine. . . . 76
3.3 A flowchart of the contact detection routine. . . . 77
3.4 A flowchart of the rigid contact routine. . . . 83
3.5 A 2D schematic of contact between the rigid impactor and a beam structure made in GeoGebra. Node N_i causes a penetration of c_i, resulting in contact force F_{ci} on node N_i and an opposite contact force F'_{ci} on the impactor. . . . 84
3.6 The partitions of the lattice for parallel computing. . . . 85
3.7 A schematic of the interpenetration of two contacting beams in time, made in GeoGebra. 89

3.8 A schematic of two contacting beams, made in GeoGebra. 90
3.9 A diagram of how a SUMMIT application works. 92

4.1 The unit insideBCC cell used throughout the test applications. Also the set-up of case I. 95
4.2 A schematic of the loading of test case I, made in GeoGebra. 96
4.3 Four simulation frames for case I. Each frame corresponds to approximately 0.04 s. . . . 97
4.4 The energy balance of the system from case I for $R_{\text{disp}} = -0.4$ and $r_{\text{disp}} = 0.005$. 98
4.5 The energy balance of the system from case I for $R_{\text{disp}} = -0.4$ and $r_{\text{disp}} = 0.05$. 99
4.6 A schematic of the loading of test case II, made in GeoGebra. 100
4.7 Four simulation frames for case II. Each frame corresponds to approximately 0.0004 s. . 101
4.8 The final state of case II for $v_0 = -0.5$ m/s. 102
4.9 The final state of case II for $v_0 = -1.0$ m/s. 102
4.10 The energy balance of the system from case II for $v_0 = -0.1$ m/s. 103
4.11 The energy balance of the system from case II for $v_0 = -0.5$ m/s. 103
4.12 The energy balance of the system from case II for $v_0 = -1.0$ m/s. 104
4.13 A schematic of a ball colliding with a wall, from the lecture notes by Richard Fitzpatrick (2006) on classical mechanics [105]. 105
4.14 The set-up for case III. The impactor unit cell is rotated 45 degrees around the z-axis compared to the stationary unit cell, and it moves downwards with 3.0 m/s. 106
4.15 Four simulation frames for case III. Each frame corresponds to approximately 4.0e-5 s. . 107
4.16 The final state for case III. 108
4.17 The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 10.0$. . . 108
4.18 The error plot for the mechanical energy from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} \in [2.5; 20.0]$. e_S is the cumulative relative error of the mechanical energy. 110
4.19 The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 20.0$. . . 111
4.20 The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 2.5$. . . . 111
4.21 Missed contact in case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 2.5$. 112
4.22 The error plot for the mechanical energy from case III for $v_0 = -3.0$ m/s and $\Delta t \in [10, 5, 1]$ ns. e_S is the cumulative relative error of the mechanical energy. Logarithmic scaling is used on the x-axis. Note that the x-axis has been inverted. 113
4.23 Four simulation frames for case IIIb. Each frame corresponds to approximately 4.0e-7 s. 115
4.24 The state for case IIIb at $t \approx 0.00240$ s. 115
4.25 The energy balance of the system from case IIIb for $v_0 = -50.0$ m/s and $R_{\text{contact}} = 1000.0$. 116
4.26 A semi-logarithmic plot of the energy balance of the system from case IIIb for $v_0 = -50.0$ m/s and $R_{\text{contact}} = 1000.0$. The y-axis has a logarithmic scale so that W_{frac} is better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable. 117

5.1 The set-up for test case 1A. The impactor is propelled in the positive z-direction. The thickness of the struts has been exaggerated for visual clarity. 122
5.2 Four simulation frames for case 1A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 123
5.3 Four simulation frames for case 1B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 124
5.4 Four simulation frames for case 2A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 125
5.5 Four simulation frames for case 2B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 126
5.6 Four simulation frames for case 3A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 127
5.7 Four simulation frames for case 3B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane. 128
5.8 The time history of the z-position of the impactor at initial velocity $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s for all cases. Note that the impactor starts off at a z-position of -0.005 m. 129
5.9 The time history of the z-velocity of the impactor at initial velocity $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s for all cases. 129

5.10 Better legible versions of the time history of the z-velocity of the impactor at initial velocity v_0 = 350.0 m/s and v_0 = 150.0 m/s for all cases. 130
5.11 The size of a single partition for case 1A. The partition is coloured in red and the thickness of the truss members has been emphasised. 131
5.12 A sectioned view of the lattice from case 1A to show that fragments can reach velocities higher than the impactor's velocity. Highlighted by red bubbles are some fragments which have been propelled into the structure following impact. Note that this sectioned view only shows a thin strip of the lattice for visibility reasons. 132
5.13 The time history of the energy components of the system at initial velocity v_0 = 350.0 m/s for cases 1A and 1B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. 133
5.14 The time history of the energy components of the system at initial velocity v_0 = 350.0 m/s for cases 1A and 1B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable. 133
5.15 The time history of the energy components of the system at initial velocity v_0 = 350.0 m/s for cases 2A and 2B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. 134
5.16 The time history of the energy components of the system at initial velocity v_0 = 350.0 m/s for cases 2A and 2B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable. 134
5.17 The time history of the energy components of the system at initial velocity v_0 = 150.0 m/s for cases 3A and 3B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. 135
5.18 The time history of the energy components of the system at initial velocity v_0 = 150.0 m/s for cases 3A and 3B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable. 136
5.19 The mass-normalised energy absorption ΔW^* for each case for an initial velocity v_0 = 350.0 m/s and v_0 = 150.0 m/s. 137

A.1 The energy plots for the other values of R_{contact} from the contact penalty sensitivity study from Subsection 4.3.2. 154
A.2 The energy plots for $\Delta t = 0.5 \cdot 10^{-8}$ s and $\Delta t = 1.0 \cdot 10^{-9}$ s from the convergence study from Subsection 4.3.3. 155

List of Tables

3.1 Overview of beam models (partially adapted from [69]) 63
3.2 Overview of the beam-to-beam contact formulations. 72

4.1 Simulation parameters for case I: the compression of a single insideBCC cell. 95
4.2 Simulation parameters for case II: the impulse loading of a single insideBCC cell. 100
4.3 Simulation parameters for case III: the collision of two insideBCC cells. 106
4.4 Simulation parameters for case IIIb: the collision of two insideBCC cells at higher velocity. 114

5.1 The properties of the three possible material choices. 119
5.2 The geometric parameters for the truss-based lattices used in the simulations. 120
5.3 General simulation parameters for the lattice impact simulation. 120
5.4 The specific input parameters used by each test case. 121
5.5 A summary of the energy absorption performance of each test case. The third column lists the relative difference in mass-normalised energy absorption performance between a case with self-contact (A) and its associated case without (B). 138

1
Introduction

Progress in the field of metamaterials in the past decades has been extraordinary. Materials once thought merely figments of human fantasy have now taken physical form, breaking through the limits imposed by nature on materials science. What sets metamaterials apart from conventional materials is their exceptional material properties unseen in regular materials, which emerge as a consequence of their structure rather than their composition. Tailoring this structure to fit specific needs allows engineering to solve problems too challenging for normal materials [1].

1.1. Architected cellular materials

Metamaterials have been growing in popularity because of the extraordinary properties they exhibit and their potential to reinforce natural properties. Extraordinary properties are extraordinary in the sense that natural materials usually do not possess such properties. Some examples are negative refractive index, negative Poisson's ratio and even negative stiffness. Examples of how metamaterials can enhance already existing properties are optical anisotropy and, for mechanical metamaterials, ultra-high stiffness/toughness/strength, ultra-low mass or high specific energy absorption [1]. Advances in manufacturing techniques, and especially in the context of additive manufacturing, have also contributed to the rise of metamaterials, which allows for realising architected materials of almost any configuration and complexity [6].

One particular property of mechanical metamaterials which is of prime relevance to this thesis, is their capacity for high specific energy absorption. This makes metamaterials quite well-suited to impact problems. Metamaterials can achieve such high energy absorption performance because they are able to approximate the ideal energy-absorbing behaviour of large volumetric decrease at constant pressure under impact. In doing so, they stretch out the duration of the impulse and keep the peak stresses below the level required to initiate damage. The way metamaterials can exhibit such behaviour is through a combination of their low relative volume density, their tendency to stiffen under compression (densification) and their tailorable mechanical properties [7] [2] [3].

Impact protection plays a crucial role in high-performance fields like the aerospace industry. Both in the skies and in orbit, protecting the structures from damage caused by collisions with objects travelling at high speeds is an on-going struggle. In space, structures are at risk of colliding with small but extremely fast moving particles. According to ESA, there are over a million objects from 1 to 10 cm estimated to be in all orbits [4]. As one can see from Figure 1.1, this number will only continue to grow as the years go on. Even at this small of a size, the particles travel at speeds well over 1 km/s, which is enough to cause major damage even if the particle is small compared to the space structure [8].

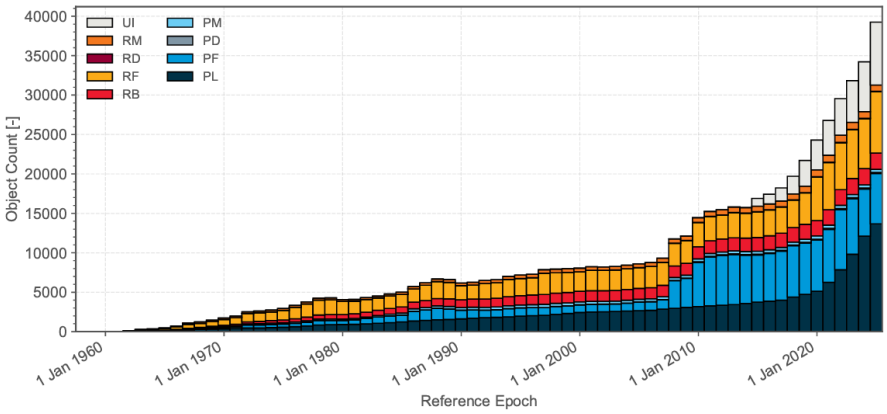


Figure 1.1: The evolution of the number of objects in all orbits, from [4]. The acronyms mean the following: UI means unidentified objects, RM is rocket mission related objects, RD means rocket debris, RF is rocket fragmentation debris, RB means rocket body, PM signifies payload related objects, PD is payload debris, PF means payload fragmentation debris, and PL is payload.

Closer to earth, aircraft also deal with foreign object debris collisions. Of course, the impact velocities are nowhere near as high as in space, but the effects of impact damage are just as adverse. Although aircraft damaged by impact events can be repaired while satellites damaged in orbit are more than likely lost or at least permanently damaged, most aircraft serve a vital commercial role. According to a study on the effect of bird strikes on aircraft repair and withdrawal costs, bird strikes alone result in almost 1 billion euros worth of damages (directly as a result of the strikes or indirectly due to delays and their repercussions) annually for the entire aviation industry [5]. So while it is true that damaged aircraft can be repaired, the loss of revenue and the direct and indirect costs of the repairs are also quite serious. This is all to highlight how important impact-resistant materials are in aerospace. Because metamaterials can outperform conventional protective materials owing to their more favourable combination of density and high specific impact resistance, they seem promising candidates for advanced protective structures [9].

1.2. Modelling impact in metamaterials

Computational models of impact in metamaterials are extremely powerful and useful tools which can serve multiple ends. From a design-synthesis point of view, computational studies facilitate the whole design process of an impact-resistant material. From feasibility studies to final testing, numerical models serve both as ways of informing design choices early on in the design, and as a means of verifying behaviour and performance late in the design cycle. Computational models can reveal early on which combinations of properties could work, lowering the risk of developing dead-ends. They can also supplement destructive testing by simulating the performance of multiple designs without incurring the costs of destroying them, allowing designers to commit to destructive testing designs they are certain will perform well. In all, being able to simulate impact and its effect on the structure accurately could save designers massively on both time and cost.

The potential gains of high-fidelity computational models of impact in metamaterials makes them worthwhile to develop, but building such a computational framework is easier said than done. Impact is a very transient and violent event, which requires a model capable of capturing the dynamic effects caused by impact. Such effects include wave propagation and fracture. Section 2.3 covers these physical phenomena and more in detail. In the context of truss metamaterials, one must also keep in mind that the models will need to describe the dynamic behaviour of each strut, including their failure and contact interactions as the lattice is deformed. The state-of-the-art computational frameworks for modelling impact in truss metamaterials is the topic of Section 2.4. It identifies the three main components of the numerical approaches from literature: the choice of element, modelling contact and representing damage. These components also invite the necessary questions: which element type is preferred, how should contact between elements be modelled, and which kind of model for damage should be used.

The literature does not have a single answer for each of these questions. Some authors use continuum elements ([10] [11] [12] [6]); others use beam elements ([13] [14] [15]). The choice of element type directly influences the contact model since beam elements would require a beam-to-beam model, which has certain implementation particularities compared to continuum elements (see Subsection 2.6.6). Lastly, if damage is represented, then the literature generally used damage evolution models to degrade the elements and eventually delete them ([10] [14] [12]).

1.3. Research approach

As stated above, the aforementioned state-of-the-art computational frameworks for modelling impact in truss metamaterials have three key constituent models: the pristine strut mechanics, the contact interactions between the struts, and the progressive damage of each strut. One thing of note is that none the works consulted in Section 2.4 use a brittle fracture model to model damage, seemingly preferring to use damage evolution. As such, there appears to be a niche currently not filled by a computational framework for impact in truss metamaterials: a beam-based model which combines beam-contact behaviour with beam-fracture behaviour. Attempting to occupy this niche with the unified beam-contact-fracture framework could provide a means of leveraging the computational efficiency of beams for slender structures while still being able to capture the complex phenomena inherent to impact such as inter-strut contact when the lattice is compressed or fracture of the struts due to the penetration of a projectile or a combination of both when fragmented struts start colliding with intact struts.

The abovementioned gap identified in the literature leads to the following research question.

Research question

How can a unified contact and fracture finite element framework improve dynamic impact simulations of beam-based truss metamaterials?

In the process of answering the research question, this thesis will execute the subtasks listed below.

(1) Incorporate beam-contact behaviour with beam-fracture behaviour, and tie it with the strut dynamics [16] [17].
(2) Test the functionality of the unified beam-contact-fracture framework in simple numerical experiments.
(3) Apply the unified beam-contact-fracture framework to a large-scale numerical model of impact in a truss metamaterial and evaluate the effect of self-contact on the energy absorption.

2

Literature review

This chapter can be divided into two halves: the first half covers the background and the state-of-the-art in modelling impact in metamaterials, while the second half treats the theory and formulations required to develop a framework for modelling impact in truss metamaterials. In concrete terms, the first leg of the literature review starts with a review of architected cellular materials (Section 2.1), followed by an examination of some works involving truss-metamaterial impact experiments in Section 2.2. After this, Section 2.3 covers the theory of impact problems with the aim of identifying the dissipation mechanisms. The first half is rounded out by Section 2.4, which analyses the state-of-the-art modelling approaches for truss-metamaterial impact problems. It is here that a gap in modelling self-contacting and fracture behaviour in beam-based struts of truss metamaterials is identified.

The second leg is dedicated to the components necessary for realising a unified contact and fracture computational framework. To this end, it opens with a review of the potential beam formulations for modelling the beams in Section 2.5, then the theory of contact mechanics along with the application to the beam-contact problem is treated in Section 2.6, and lastly, Section 2.7 covers the fracture model of choice when modelling beam fracture. In all, through the theoretical and practical background, and by covering the building blocks for a computational framework for impact in truss metamaterials, this literature review serves the final objective of modelling a large scale truss-metamaterial impact problem which captures both fracture and self-contact.

2.1. Architected cellular materials

Architected cellular materials have emerged as a topic of extensive research in materials science. Such materials have the vaunted properties of being able to sustain large deformations while being able to approximately return to their original shape after unloading. In nature, cellular materials generally possess irregular cell structures, making their geometric configurations difficult to reproduce in engineering applications. However, with the advances made in the field of additive manufacturing, creating cellular structures of nearly any complexity is no longer out of reach [6].

2.1.1. Metamaterial classification

There is no one way to classify a family of materials as broad and customisable as metamaterials. One possible method of classification is based on bulk material behaviour, which also ties into their use. Exact taxonomy aside, the literature generally agrees on the following distinct classes of metamaterials: acoustic, electromagnetic, mechanical and thermal [18] [19].

Figure 2.1 and Figure 2.2 give a good indication of not only the different metamaterial classes, but also the sheer breadth of their applications.

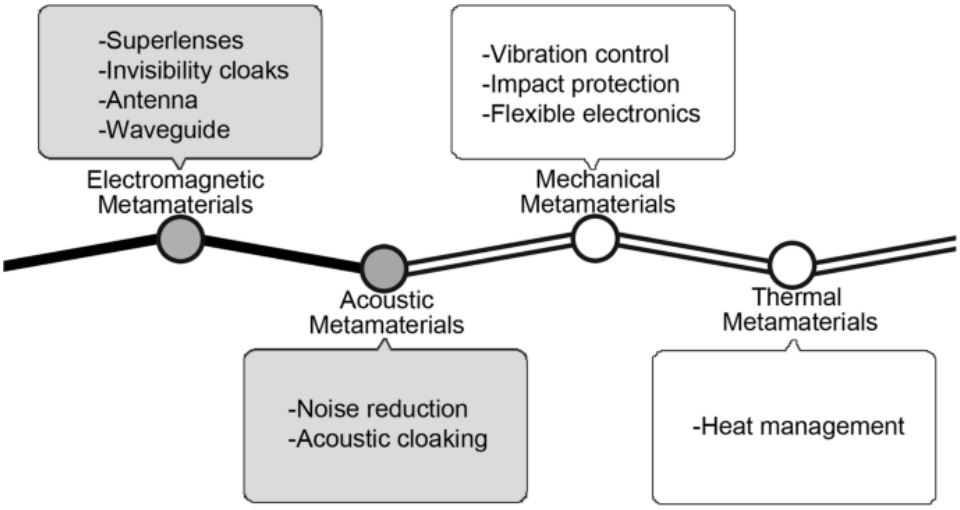


Figure 2.1: The different types of metamaterials and the associated area of application, from [18].

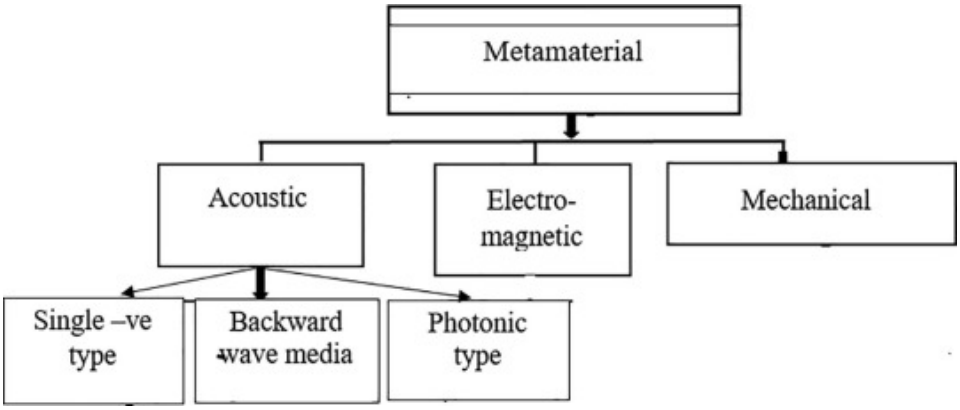


Figure 2.2: The different types of metamaterials according to Kumar et al. (2022) [19].

One could go on at length over the different types of metamaterials and their uses; however, the specific class of metamaterials most relevant to this thesis are the mechanical metamaterials. As such, the remainder of this literature review shall concern itself solely with mechanical metamaterials. That said, it cannot be overstated how widely applicable metamaterials are outside of mechanical design.

Because the properties of metamaterials are governed mostly by structure, rather than composition, one could also classify metamaterials based on their cell structure, Figure 2.3.

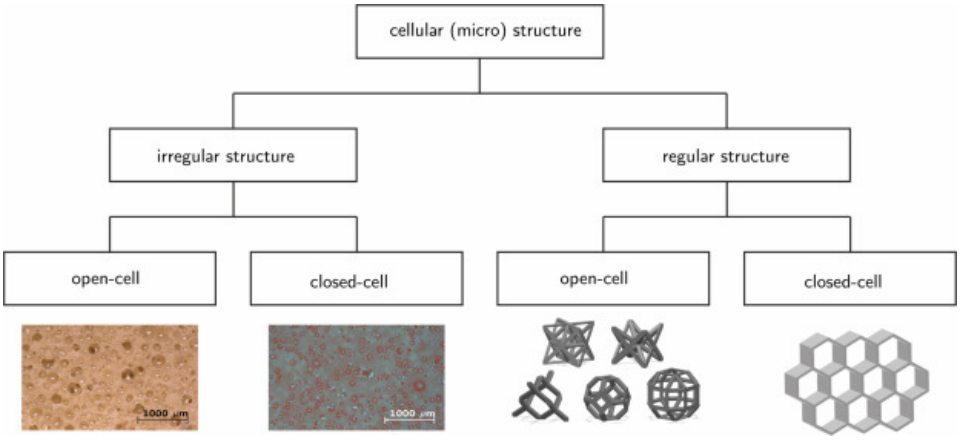


Figure 2.3: A classification of cellular materials based on structure, from [6].

Cellular materials either have an irregular structure or regularly repeating units. Both irregular and regular structures can be further divided depending on whether the structure is open-celled or closed-celled. In irregular structures, the cell or pore size is usually gradually or stochastically distributed, like in bark, cork or foam. Closed-cell foams generally have spherical pores, as opposed to open-cell foams, where the pores grow into each other. This is why the ligament structure describes open-cell foams better [20] [21].

In the case of regular structures, closed-cell configurations are mostly created by extending the two-dimensional structure in the third dimension, analogous to how the honeycomb core in sandwich structures is produced. Meanwhile, open-cell regular structures comprise interconnected struts arranged in a three-dimensional space. An example of open-celled metamaterials are lattice materials, which resemble crystal lattices and consist of lattice elements such as beams or shells. The connectivity of these lattices directly determines the overall mechanical behaviour [6].

It is these crystal lattice-based architected materials which are of particular interest because of the existing body of work examining their structural properties and manufacturing methods [22] [23] [24] [25], and the relative simplicity of modelling.

2.1.2. Crystal lattice metamaterials

Lattice metamaterials have a periodic, porous structure that consists of basic units assembled in some periodic combination [26]. Following the structural taxonomy of cellular materials described earlier, this means lattice metamaterials are regularly structured cellular materials. Therefore, lattice metamaterials can be divided into 2D and 3D lattice metamaterials, depending on their structural configuration, see Figure 2.4.

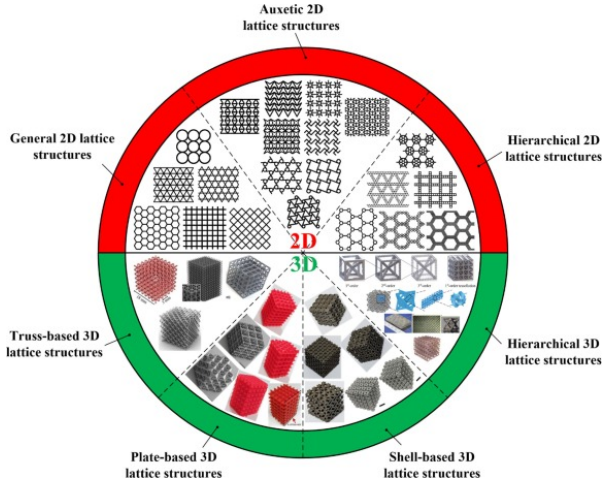


Figure 2.4: The different structural configurations of lattice structures, from [26].

The 2D lattice structures primarily consist of honeycomb structures, auxetic structures and hierarchical 2D lattice structures [26]. Honeycomb structures have been extensively used in lightweight structural applications, particularly in aerospace [27]. Auxetic structures show negative Poisson's ratio effects when subjected to load [26], which makes them a prime subject of study [28] [29] [30]. Hierarchical 2D lattice structures have unit cells which include some form of substructure. This results in multi-level characteristics, similar to natural structures like bone [26]. This kind of 2D lattice metamaterial has also been the topic of many studies [31] [32] [33].

Meanwhile, 3D lattice structures can be roughly grouped into truss-based, plate/shell-based and hierarchical 3D lattice structures. Truss-based lattice structures consist of unit cells constructed from interconnecting truss members and assembled in all manner of configurations. Currently, they are the most widely studied 3D lattice structures [26]. The large variety of possible configurations has lent itself to an extensive body of work [34] [35] [36] [37] [38] [14]. Plate- and shell-based lattice structures often comprise unit cells with similar topology to those used in truss-lattice structures, but assembled with plates or shells [26]. Many studies have been carried out to find out how using plates or shells can exceed the performance of truss-lattice structures [39] [40] [41] [42]. Lastly, hierarchical 3D lattice structures have multi-level characteristics because of a combination of substructures in their unit cells, similar to 2D hierarchical lattice structures [26]. Their potential to imitate natural structures has brought about many studies [43] [44] [45].

Although crystal lattices can be constructed from all three types of 3D lattice structures mentioned earlier, special attention will be given to the truss-based lattice structures. Despite the works of Tancogne-Dejean et al. (2018) and Bonetti et al. (2019) showing that plate- and shell-based lattice structures can outperform even the best truss-based lattice structures [39] [42], truss-lattice metamaterials are the metamaterial of choice for this thesis because of their relative modelling simplicity, their compatibility with the numerical framework used by this thesis, and their familiarity within the department of Aerospace Engineering at Delft University of Technology, with recent works related to truss-based architected materials [46] [47] [48].

As mentioned before, there are many different cell topologies used to assemble truss lattices. Before the geometry of specific configurations is examined, it is worthwhile to first define the unit cell. To do so, one must draw upon the theory of crystallography. An ideal crystal consists of an infinite repetition of identical groups of atoms. This group is referred to as the basis, and the set of mathematical points to which this basis is attached is the lattice. The way to identify the unit cell is through the following steps: first, draw lines from a lattice point, i.e. a node, to all nearby lattice points; then, at the midpoints of those lines, draw new lines or planes orthogonal to the previous lines. The smallest volume enclosed by the new lines or planes is known as the Wigner-Seitz cell [49], Figure 2.5.

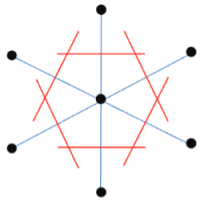


Figure 2.5: The construction of a Wigner-Seitz cell, from [49].

The Wigner-Seitz cell can have an arbitrary combination of edge lengths and may not always have right angles between edges. However, for simplicity, one can imagine a unit cell being a cube with possible node locations being the vertices, the centre of the faces or the centre of the cube [6], see Figure 2.6.

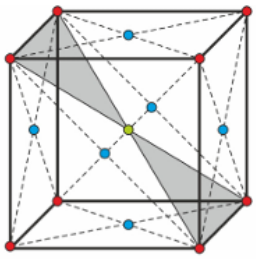


Figure 2.6: A unit cell and its possible node positions, from [6].

Having established an image of the unit cell, it is now easier to describe the geometry of specific cell configurations. Some common examples of unit cell configurations are the simple cube (SC), the body-centred cube (BCC), the face-centred cube (FCC), the octet, and the octahedron [26]. The simple cube is a unit cell with nodes only at the vertices [49]. The BCC unit cell features an extra node at the centre of the cube. Both have been illustrated in Figure 2.7.

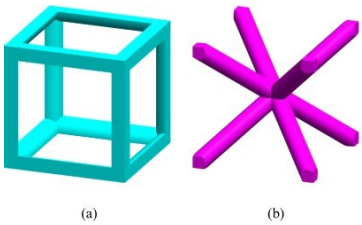


Figure 2.7: The simple cubic cell (a) and the body-centred cubic cell (b), from [34].

The FCC unit cell has nodes at the vertices and at the centres of each face [49], Figure 2.8. The octet cell is composed of two Platonic solids: the tetrahedron and the octahedron, and it resembles the FCC cell [50]. The octet cell can be seen in Figure 2.9.

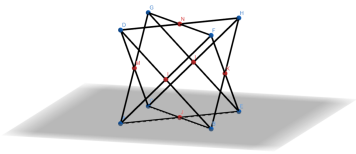


Figure 2.8: The face-centred cubic cell, made in GeoGebra.

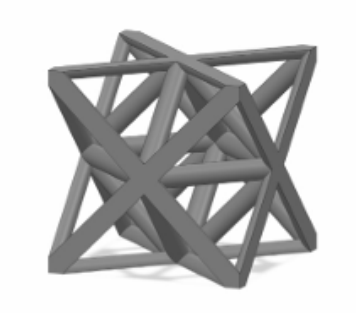


Figure 2.9: The octet unit cell, from [6].

The octahedron unit cell has nodes only at the centres of the faces of the cube [38], see Figure 2.10. If the octahedron is larger than the cube, and the vertices that stick out of the cube are truncated, then one ends up with a truncated octahedron [6], Figure 2.11.

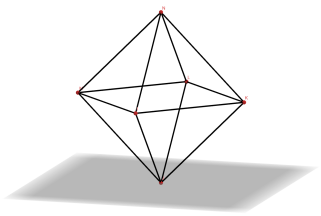


Figure 2.10: The octahedron unit cell, made in GeoGebra.

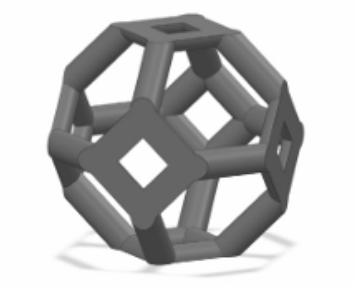


Figure 2.11: The truncated octahedron unit cell, from [6].

Intuitively speaking, one may think that the configuration has an effect on the mechanical behaviour of the lattice. One way to quantify this effect is through the Maxwell number, M. Without going too much into the details, the Maxwell number depends on the combination of nodes and the number of struts connected to the nodes. The value of the Maxwell number then dictates whether the structure has stretch-dominated behaviour ($M \geq 0$) or bending-dominated behaviour ($M < 0$) [6].

The fact that different configurations result in different mechanical behaviour presents an opportunity for designers to tailor the lattice to the problem. This means that one can fine-tune the mechanical properties with the same constituent materials by changing the unit cell topology.

2.1.3. Mechanical properties

When a mechanical property of a material such as yield strength or Young's modulus is plotted against its density, the vast majority of commercially available materials end up on the main diagonal of the

plot [2], see Figure 2.12.

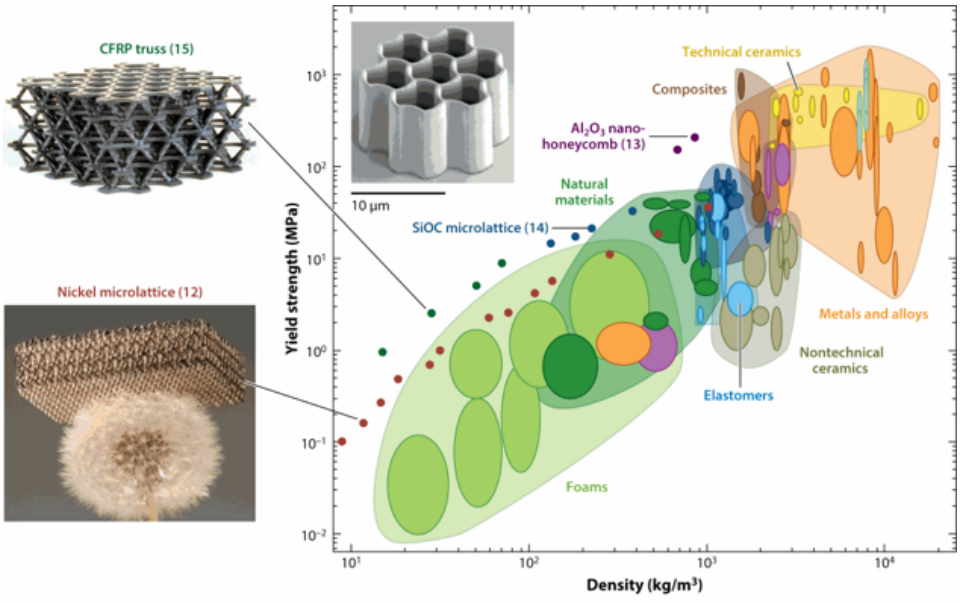


Figure 2.12: An Ashby plot which shows white regions unoccupied by commercial materials, and some cellular materials that can exist in the white regions, from [2].

As one can see, this ends up creating voids in the Ashby plot, wherein no material is available that has this combination of mechanical property and density. Although there exists a void both above and below the main diagonal, the empty space above is naturally of greater engineering interest. Particularly in the low density range, the options are sparse: foams, honeycombs and some natural materials.

However, owing to the tailorability of architected materials, it becomes possible to push into the white regions. Because the bulk material behaviour is also dictated by the cellular architecture of such materials, and not just their constituents, it allows designers much greater control over mechanical properties by varying the solid volume fraction of the material and the spatial configuration of the lattice. If one desires a low-density material, one can use a small solid volume fraction [2]. Designers can then optimise the lattice configuration for this volume fraction to get the most out of the architected material.

In a previous work by Schaedler et al. (2011), nickel microlattices are manufactured which access the white region across a wide range of density, modulus and strength [51]. Other examples of architected materials extending the material design space into the void are alumina nanohoneycombs [52], ceramic silicon oxycarbide microlattices [53], and carbon fibre-reinforced polymer truss structures [54] [55]. These materials can also be seen in Figure 2.12.

2.1.4. Applications of mechanical metamaterials

Even when the scope is narrowed down to mechanical metamaterials only, there are still numerous fields of application where this subset of metamaterials has immense potential. This has to do with the many ways mechanical metamaterials can be realised. Starting from fundamental material mechanics, one can further subdivide mechanical metamaterials into lightweight metamaterials, characterised by a very high specific stiffness; pattern transformation metamaterials, which have a switchable stiffness; negative compressibility metamaterials; pentamode metamaterials, known for having fluid properties despite being solids; and auxetic metamaterials, famed for having a negative Poisson's ratio [56].

Given the many combinations of cellular configuration and the degree of customisability of architected materials, it may be easier to list the applications mechanical metamaterials are not suited for. Their applications range from structural to functional applications such as electronics and biomedical engi-

neering. In the structural field, they can be tailored to have ultra-high strength, form ceramics which do not fail catastrophically when overloaded, obtain extremely high recovery after loading, and even enable mechanical cloaking, all at low density [57].

When mechanical metamaterials are intended for use in electronics, they can potentially provide an optimal balance between high charge-discharge rates and mechanical robustness, allowing them to act as energy storage devices under operating conditions [57]. In the biomedical field, mechanical metamaterials can be leveraged to promote organised cell growth as biological scaffolds, act as constructs for tissue, serve as deployable angioplasty or oesophageal stents, or even be used in artificial implants with tuneable mechanical properties [57] [56].

The literature makes it clear that metamaterials have an extremely wide range of applications, but the most relevant application field for this thesis is the structural field. Specifically, the structural application of interest is impact protection. Architected cellular materials are extremely well suited for this application because of their energy absorption performance [2] [7]. Their ability to absorb energy is highly dependent on geometrical configurations, deformation mechanisms, material failure modes and loading conditions. This allows designers more freedom to tailor the response to impulsive loads compared to conventional materials [7] [2].

According to Jiang et al. (2022), an energy-absorbing structure usually fails due to uniaxial compression loading, and the associated load-displacement curve shows a near-constant stress level from the initial yield point until the onset of densification. An ideal energy-absorbing structure should have a prolonged stress plateau with peak stresses beneath the damage threshold [7]. To add to this, a protective medium must be able to undergo a large volume decrease at constant pressure, so that the incoming pressure is lowered below the damage threshold, and the duration of the impulse is extended [3] [2].

These requirements necessitate a stress-strain response with maximum Young's modulus, a constant crushing stress that does not exceed the damage threshold, and a maximum densification strain. Dense solids and fluids are unsuitable since they are nearly incompressible [2]. However, architected cellular materials can offer potential solutions through a combination of relative volume density, densification strain and tailored mechanical properties [3].

Impact protection plays a central role in industries such as the automotive industry, the defence industry and the aerospace industry. When subjected to transient dynamic loading, materials must deal with stress waves that rapidly propagate through them. These stress waves often exceed the material's elastic limit, which results in irreversible dynamic damage mechanisms like plastic deformation, crack growth and eventual structural failure [58]. Especially in high-performance sectors such as aerospace, well known for having an unrelenting requirement for lightweight and reliable design, there is a more pronounced demand for advanced protective materials. The problem is that conventional protective materials usually cannot achieve high impact resistance and low density at the same time [9].

Because architected cellular materials have the potential to rise to the demands of a lightweight impact-resistant structure, this has spurred a multitude of studies into metamaterials and their application in impact protection. Recent works in rod/beam-based metamaterials include a bio-inspired snap-through metamaterial using hybrid stretching- and bending-dominated designs to achieve ultra-high load-bearing capacity and significant energy dissipation at the same time [59]; a snap-fit-based modular multi-stable metamaterial which uses sequential snap-through to create multi-stage energy absorption and reduce peak forces [60]; a study of additively manufactured elastomer lattices that revealed topology-driven behaviour concerning stretching-dominated and bending-dominated layouts [61]; an investigation of artificial grain-boundary hardening of lattice-based architected materials [62]; a characterisation of elastic performance of hollow truss lattices with various configurations [63]. These research efforts capture the many ways in which architected materials can be tailored to maximise their impact resistance for their intended use cases.

2.1.5. Summary of the architected cellular materials review

Metamaterials come in all forms and have many different uses. One can categorise them based on their use, such as mechanical, electromagnetic or acoustic. However, in the case of cellular materials, one can also classify them based on their topology, such as whether they are regularly or irregularly structured, and whether each unit cell is open or closed.

One specific type of cellular material, the truss-based lattice material, is of particular interest because their mechanical properties are well documented, and they are relatively simple to model. Because the cell configuration largely dictates the mechanical behaviour of the lattice material, it allows designers to fine-tune the mechanical properties of the lattice material to the problem.

This tailorability of architected materials enables engineers to explore niches currently not occupied by commercial materials. One such niche that is especially of interest to many industries, is that of high Young's modulus or strength and low density. Conventional materials at the low density end of the material chart do not have high mechanical properties. However, metamaterials can be designed to have low density and relatively high mechanical properties.

Since metamaterials can obtain high mechanical properties per unit weight, this makes them quite interesting candidates as materials for advanced lightweight structures. Particularly, there is much work on metamaterials as impact-resistant structures. High-performance industries such as aerospace, where structures must balance reliability with lightweight design, especially have interest in the use of metamaterials for impact protection.

2.2. Impact experiments of truss-lattice metamaterials

As mentioned in Subsection 2.1.4, the performance of architected cellular materials as impact protection has been the subject of many studies as of late. Authors have investigated different velocity regimes, architectures and other creative strategies for optimising energy absorption. Subsection 2.1.4 listed some of the more unique studies, however, this thesis will limit the scope of architected cellular materials to ordinary truss-based metamaterials. Doing so simplifies the modelling process since the metamaterials are represented by a network of joints and struts. The struts can then be modelled using a finite element representation of choice, be it beam elements or continuum elements. In spite of the relatively simpler modelling process of truss-lattice metamaterials, the results from investigating their impact performance are not less insightful.

To facilitate the interpretation of numerical results obtained with a truss-based metamaterial, it is only fitting that some studies involving impact tests with truss metamaterials are presented. Portela et al. (2021) is about supersonic impact testing of nanoarchitected carbon materials [64], see Subsection 2.2.1. Their work marks one of the first ventures into supersonic impact testing of 3D architected cellular materials, and they investigate tetrakaidecahedron (or truncated octahedron) unit cells. Butruille et al. (2024) investigate the dissipation mechanisms of 3D architected materials in the dynamic impact regime [12], see Subsection 2.2.2. They study the octet unit cell and the tetrakaidecahedron unit cell. Bieler and Weinberg (2025) present the energy absorption of different additively manufactured unit cell configurations under impact load [6], see Subsection 2.2.3. The cell topologies studied include the octet, the BFCC (a combination of a BCC and FCC cell) and the truncated octahedron.

In the following, the studies mentioned earlier are summarised, their experimental settings described and their results presented.

2.2.1. Portela, Edwards, Veysset, Sun, Nelson, Kochmann and Greer

The topic of the paper by Portela et al. (2021) is supersonic impact resilience of nanoarchitected carbon. They investigate the impact energy dissipation of nanoarchitected carbon materials, and with supersonic microparticle impact experiments, they show that said architected materials have superior specific energy dissipation to traditional impact-resistant materials such as steel or Kevlar. Using in-situ ultra-high-speed imaging and post-mortem confocal microscopy, they discover consistent mechanisms that allow the nanoarchitected carbon materials to outperform other impact-resistant materials [64].

Experimental set-up
Portela et al. fabricated, tested and analysed 3D nanoarchitected carbon lattice materials assembled from 13500 unit cells with beam diameters ranging from 360 nm to 530 nm, and relative densities ($\bar{\rho}$) ranging from 14% to 23% subjected to impact by supersonic microparticles. Their unit cell topology of choice is the tetrakaidecahedron (truncated octahedron), because of its bending-dominated response. For a visual of this unit cell, see Figure 2.11. Each sample is comprised of a sufficiently large stacking of 30 x 30 x 15 unit cells so as to reach proper separation of scale between unit cell size and overall sample dimensions.

To investigate how $\bar{\rho}$ can influence the impact response, the authors of [64] manufactured samples of two relative densities 14% ± 2% and 23% ± 3%. This was achieved by varying the strut diameter between 370 ± 40 nm and 530 ± 40 nm, while keeping the unit cell size at 2.5 ± 0.2 μm. These nanoscale strut dimensions mean that the lattice structures fall within the size-effect regime of pyrolytic carbon, the material of choice in [64].

The microparticle impact experiments were carried out using the LIPIT (laser-induced particle impact test) method, where a 14 μm diameter SiO_2 particle is accelerated to velocities between 30 m/s and 1.2 km/s by a pulsed laser to collide with the specimen. The impact event was captured using an ultra-high framerate camera, meanwhile scanning electron microscopy and laser-confocal microscopy was used post-mortem on the specimen. The experimental set-up has been illustrated in Figure 2.13.

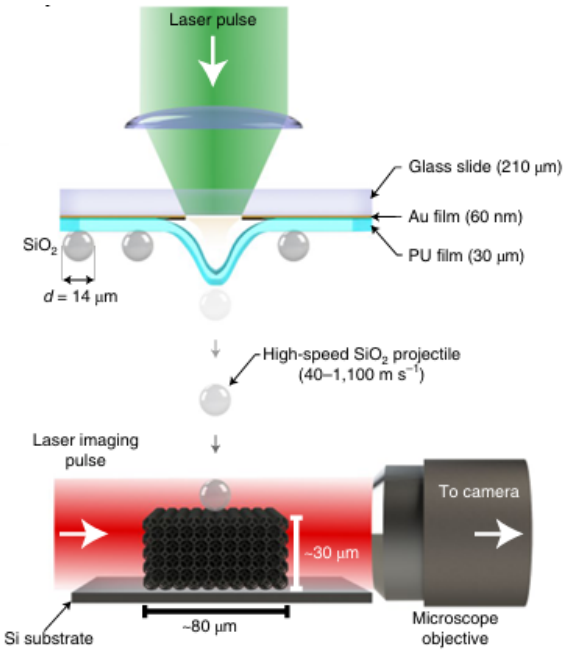


Figure 2.13: A schematic of the test set-up showing the LIPIT, the specimen and the camera, from [64].

Results
Portela et al. experimentally demonstrate that the SiO_2 particle impacts onto the specimen results in three distinct qualitative response modes: elastic impact, cratering with particle rejection and cratering with particle capture. They found that these response modes depended on the impact energy and the relative density of the metamaterial, see Figure 2.14. One can also argue that full specimen perforation is a fourth mode. When the craters were studied close-up, the authors of [64] found indications of brittle failure in the carbon struts and compaction of debris.

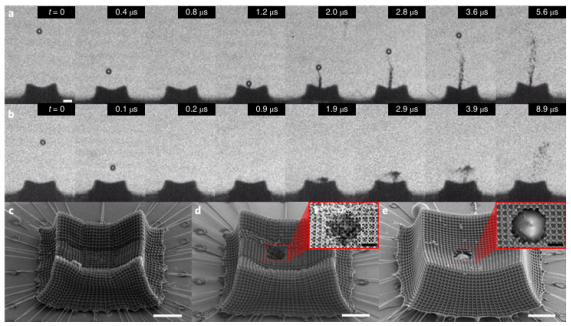


Figure 2.14: The impact responses and the resulting damage in the specimen with $\bar{\rho} = 23\%$. (a) shows impact and rebound while (b) illustrates particle capture. (c) is a sample after elastic impact. (d) is a post-impact micrograph of the sample from (a) which shows cratering, and (e) is a micrograph of the sample from (b) with the captured particle visible. The white scale bar equates to 20 μm, while the black scale bar is 5 μm. This figure can also be found in [64].

To quantify the energy absorption ability of the nanoarchitected carbon specimen, Portela et al. defined the inelastic energy due to the lattice material only as Equation 2.1.

$$W_i = W_0 - W_r - W_{i,\mathrm{SiO_2}}(W_r) \tag{2.1}$$

Here, W_0 is the impact energy, defined as $W_0 = \frac{1}{2}mv_0^2$ with v_0 the impact velocity. W_r is the rebound energy, defined as $W_r = \frac{1}{2}mv_r^2$ with v_r the rebound velocity. In the equation for both W_0 and W_r, m is the particle mass, calculated with $m = \frac{4}{3}\rho_p \pi r^3$ with particle density $\rho_p = 1850$ kg/m^3 and radius $r = 14$ μm. $W_{i,\mathrm{SiO_2}}$ is the inelastic energy contribution from densification in the microparticles, which is a function of W_r. Because calibration impact experiments showed that dissipation in the microparticles contributed to the total inelastic energy, this contribution must be decoupled from the metamaterial response.

When the energy is plotted for both relative densities, the authors obtain the results shown in Figure 2.15.

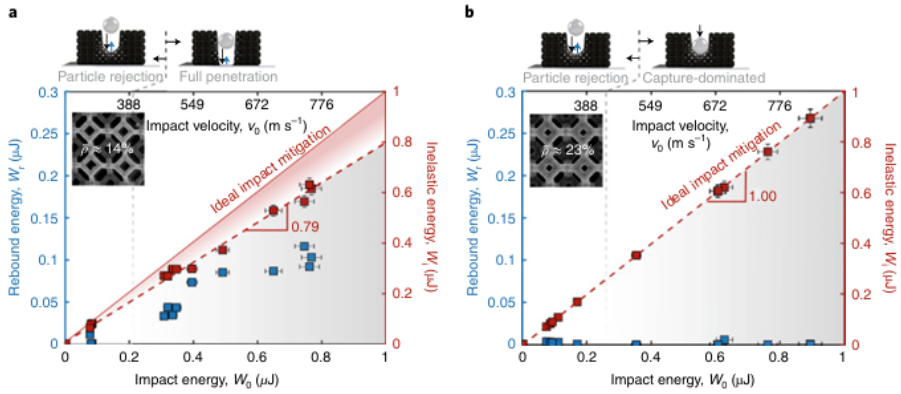


Figure 2.15: The impact energies for the experiments. (a) shows the results for the specimen with $\bar{\rho} = 14\% \pm 3\%$, and (b) is for the specimen with $\bar{\rho} = 23\% \pm 4\%$. A linear fit of the inelastic energy W_i as a function of W_0 can be seen as the red dashed lines. [64]

Portela et al. show that for low velocities (approximately 50 m/s and lower), both specimen types have an elastic response (Figure 2.14). At impact velocities above 50 m/s, the specimen with $\bar{\rho} \approx 14\%$ displayed cratering with particle rejection, and if the impact velocities exceeded 490 m/s, perforation of the specimen was observed. Meanwhile, the $\bar{\rho} \approx 23\%$ specimen show cratering and rebound behaviour beyond the elastic regime. When the impact velocity is increased to 515, their response transitions to cratering and capture behaviour.

The authors of [64] then calculated the specific inelastic energy for the purpose of comparing the impact performance of the specimen with other ballistic materials. They defined the specific inelastic energy

as $W_i^* = W_i/m_p$. m_p signifies the mass of the material that participated in the impact, that is to say, the mass of the material permanently deformed by the impact. The specific inelastic energy for both specimen types was plotted for the range of impact velocities studied, along with the specific inelastic energy for other materials used for impact-protection, see Figure 2.16.

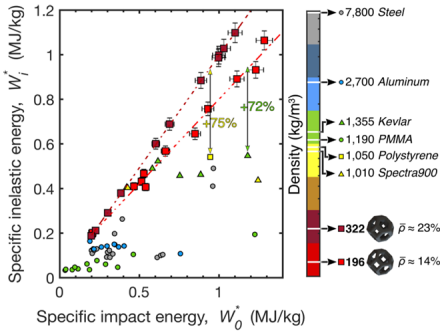


Figure 2.16: A comparison of the mass-normalised inelastic energy W_i^* as a function of v_0 with other ballistic materials, from [64].

Portela et al. demonstrate experimentally that for the same specific impact energy $W_0^* = W_0/m_p$, the nanoarchitected carbon has a roughly 70% higher specific energy dissipation than Kevlar composites and nanoscale polystyrene. They conclude that their work shows that the superior impact efficiency of nanoarchitected materials can be derived from the additional dissipation mechanisms emerging from said materials' ability to transmit compaction shocks. That said, they point out that further study is required to investigate the how the architecture of a lattice influences the cratering response.

2.2.2. Butruille, Crone and Portela

Butruille et al. investigate the dissipation mechanisms of 3D architected materials in the dynamic impact regime. They aim to decouple the contribution of each mechanism and to provide guiding principles for the design of micro-architected materials for performance in the dynamic impact regime. To this end, they have developed a microscale suspended-plate impact testing framework where supersonic microparticles are accelerated to velocities of 850 m/s against a substrate-decoupled architected material. This test is performed on two types of cell configurations, as well as the non-architected monolithic constituent material. Analysis of energy dissipation characteristics is done with help from in-situ ultra-high-speed imaging and post-mortem characterisation [12]. In a way, this study can be viewed as a follow up to [64] since this time, they do investigate multiple architectures.

Experimental set-up

Butruille et al. make a point of isolating the specimen from the substrate, which they explain was a limitation of [64]. To this end, they designed the microlattice metamaterials in a suspended form factor with two monolithic supports on either side. This has also been illustrated in Figure 2.17.

The monolithic supports are 180 μm apart, and on top of them, there is a 2D base which functions as a foundation for the specimen. Three types of specimen were manufactured: octet microlattices, tetrakaidecahedron (truncated octahedron) microlattices and monolithic polymer plates. The relative density $\bar{\rho}$ of the lattices was $33 \pm 1.2\%$ with unit cell size 6.7 ± 0.1 μm. The fabricated thickness of the monolithic plates was 7.6 ± 1.8 μm so that they were approximately mass-equivalent to the lattices.

In order to decouple the architected materials from the substrate and the lateral boundary conditions as much as possible, Butruille et al. made use of computational Bloch-wave analysis to find the dimensions of the suspended sample which minimised the effect of the boundary conditions on their impact experiments. From this, they ended up with metamaterials consisting of 32 x 26 x 5 tessellations of unit cells, which result in an approximately 214 x 174 x 33.5 μm plate of architected material.

Just as Portela et al. (2021) [64], the authors of [12] employ the LIPIT method to conduct the microparticle impact experiments. The specimen are hit only in their z, or height direction. 33 ± 2 μm diameter SiO_2

microparticles were selected as the impactors so as to maintain separation of scales between the unit cell dimensions and the impactor. These microparticles were accelerated towards the centre of the specimen at velocities ranging from 300 to 850 m/s.

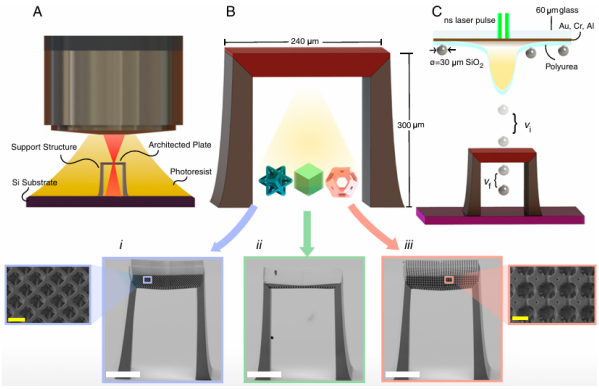


Figure 2.17: The suspended microparticle impact framework described by Butruille et al. (2024). (A) shows the fabrication of the suspended metamaterial structure using photon lithography. The result of this can be seen in (B). (B i, ii and iii) are SEM micrographs the octet, monolithic and tetrakaidecahedron structures, respectively. The white scale bars indicate 100 μm and the yellow scale bars are 5 μm. (C) is a schematic of the impact experiments with the LIPIT method. This figure can also be found in [12].

Results

The velocities of the particles could be measured before and after penetrating the specimen by using an ultra-high-speed camera, see A, B and C of Figure 2.18. Additionally, SEM allowed for post-mortem characterisation, such as identifying craters, pointing to localised cylindrical craters in the metamaterial specimens as opposed to radial-crack formation and gouging in the monolithic specimen. The SEM images can be seen in D, E and F of Figure 2.18.

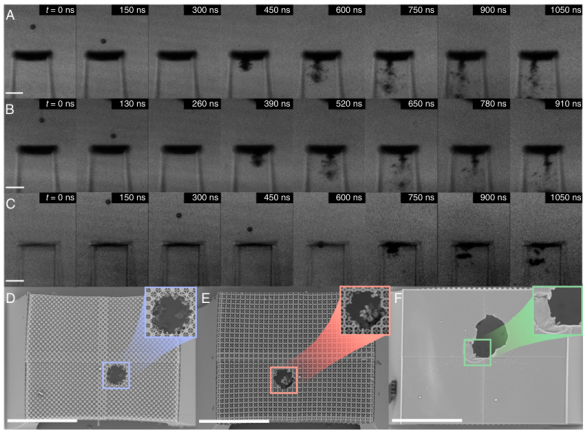


Figure 2.18: Micrographs of the impact experiments. (A) is for the octet structure, (B) is for the tetrakaidecahedron structure and (C) is for the monolithic sample. (D), (E) and (F) are post-impact SEM micrographs for the octet, tetrakaidecahedron and monolithic sample, respectively. [12]

When the measurements of the pre- and post-impact velocities are taken from the captured micrographs above, and plotted (A and B of Figure 2.19), a qualitatively similar relation can be observed for all specimens tested. The authors of [12] then extend their analysis to energy. They defined the energy dissipated by the materials as $W_{\text{dis}} = \frac{1}{2} m_{\text{sphere}}(v_i^2 - v_f^2)$. Here, m_{sphere} is the mass of the impactor, v_i is the impact velocity, and v_f is the post-impact velocity, see A and B of Figure 2.19.

To quantify the dissipated energy per unit mass, Butruille et al. normalised the dissipated energy by the crater mass of the specimen, as illustrated in C and D of Figure 2.19. This results in the metric $\bar{W}^* = \frac{W_{\text{dis}}}{\rho A_c t}$, where ρ is the mass density of the specimen, A_c is the cross-sectional area of the crater, and t is the thickness of the specimen. Normalising the dissipated energy makes it possible to compare the impact efficiency from different specimens and experimental settings.

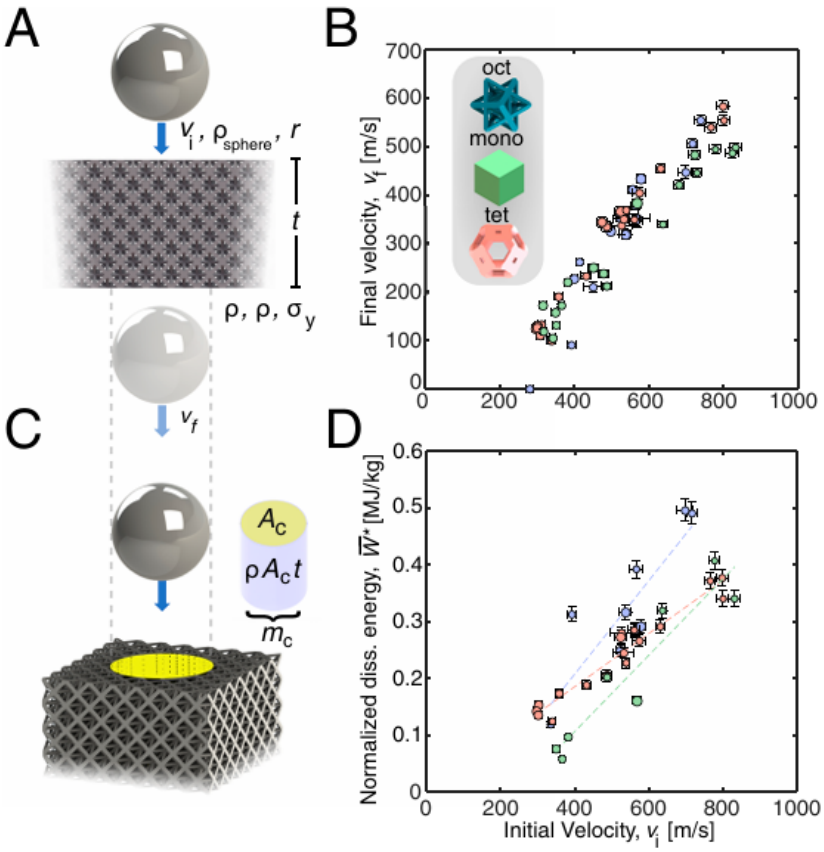


Figure 2.19: The results of the microparticle impact tests. (A) shows the quantities that can be measured. (B) plots the final impact velocity to the initial impact velocity. (C) is a schematic of how the crater mass is calculated to normalise the dissipated energy. (D) plots the mass-normalised dissipated energy against the impact velocity. These results can also be found in [12].

Although the monolithic plates were observed to dissipate the most energy of 6.00 μJ at impact velocities of 820 m/s, their impact response is less efficient as a result of radial cracks and gouging. This is in contrast to the well-localised 1D cratering mechanism which is prevalent in the architected specimens. In fact, according to the specific dissipated energy metric, the octet had the highest energy dissipation efficiency over all impact velocities, reaching a maximum value of 0.49 MJ/kg for an impact velocity of 715 m/s. The monolithic plate is generally the least efficient, but it was observed to match the performance of the tetrakaidecahedron at velocities above 800 m/s, as can be seen in Figure 2.19 D.

Decoupling dissipation mechanisms

Decoupling the dissipation mechanisms is the focal point of [12], so it was deemed fitting to separate this part of their paper from the rest of their results. Butruille et al. argue that the aforementioned results have proved effective when it comes to quantifying metrics of absolute energy dissipation, but analysis of high-speed imaging alone does not enable the identification of energy dissipation mechanisms. They expect that the dissipated energy consists of dynamic compaction contributions W_c, inertial contributions resulting from imparting kinetic energy onto the ejecta W_{in}, and contributions

from fracture boundary formation in the craters W_f. This has been illustrated in Figure 2.20 A. They then assume that the total energy dissipated can be approximated as the sum of each contribution:

$$W_{tot} \approx W_c + W_{in} + W_f \tag{2.2}$$

The authors focus on decoupling the contributions of compaction and fracture, assuming that the kinetic energy imparted onto the ejecta in their specimen remains approximately the same for impact events featuring equivalent particle size, specimen mass and velocity.

The first step Butruille et al. took in order to study W_c was to conduct quasi-static uniaxial compaction tests on their tested architectures. Substrate-bound architected materials consisting of 10 x 10 x 10 tessellations of unit cells with the same parameters as tested under impact were then subjected to compaction tests. In the tests, they used a displacement-controlled nanoindenter with a strain rate of $\dot{\epsilon} = 10^{-3}$ s^{-1}. When these experiments for both types of architectures are conducted until just before densification, at $\epsilon \approx 50\%$, then metrics of effective Young's modulus E^*, yield strength σ_y, and dissipated compaction energy density, which they defined as the area under the stress-strain curve, can be recovered.

The authors report that the tetrakaidecahedron reached a mass-normalised energy dissipation of 30.0 ± 1.40 kJ/kg (C and E of Figure 2.20), while the octet managed a mass-normalised energy dissipation of 19.08 ± 2.79 kJ/kg (B and D of Figure 2.20). The observed results for quasi-static compaction is the opposite of the observed impact performance. Butruille et al. then conducted high-fidelity finite element simulations on the architected materials with the goal of better understanding the mechanisms that go into the observed compaction response. The results of the numerical simulations can also be seen in D and E of Figure 2.20. Overall, one can see that the simulated response generally matches the experimental results.

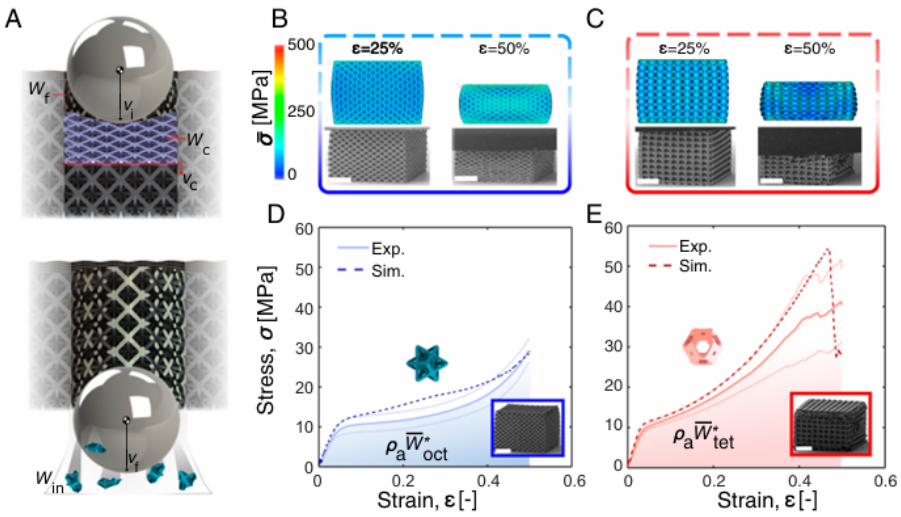


Figure 2.20: The decoupled compaction mechanism of microlattices. (A) is a schematic of the energy dissipation components during impact, showing W_f due to fracture, W_c due to compaction and W_{in} due to inertial effects. (B) and (C) show the simulated and experimental compaction behaviours of an octet and tetrakaidecahedron lattice, respectively. (D) and (E) depict the simulated and experimental stress-strain curves for both lattices. The scale bar equates to 20 μm. [12]

The authors then treated their method of investigating the contribution of fracture to the total dissipated energy. To this end, they devised a quasi-static coupled compaction-fracture experiment to characterise the energy required to cause a cylindrical fracture surface in an architected microlattice. This experiment attempts to mimic the cratering caused by the microparticles in the impact experiments. The test involves quasi-statically pushing a monolithic cylindrical pillar into a microlattice. Doing so would create a fracture surface around the perimeter of the pillar, and it would result in localised compaction beneath the pillar. The results of this can be seen in Figure 2.21.

The specimens for the pillar indentation test were octet and tetrakaidecahedron lattices comprising an assembly of 18 x 18 x 14 unit cells. The parameters were still the same as used in the impact tests. Mounted on top of the lattices are monolithic polymer micropillars with diameters 25, 30 and 35 μm. During the experiment, the pillars were uniaxially forced into the lattice at a strain rate of $\dot{\epsilon} = 10^{-3}$ s^{-1}. At the same time, the force-displacement (F-δ) response of each test was saved. The authors explain that different diameter pillars were intended to investigate the effect of impactor particle size, and comparing the results of this experiment between the two architecture types could reveal how the architecture influences the energy dissipation.

With the height H of the lattices, and the cross-sectional area A of the pillars (A, i of Figure 2.21), the force-displacement results were turned into stress against normalised displacement (F/A against δ/H) responses. The mass-normalised energy dissipation $\bar{W}^*$ for each fracture experiment was then calculated by integrating the aforementioned relation. The results were plotted in Figure 2.21 D. The authors observed that the octet lattice specimen, in spite of exhibiting a lower specific energy dissipation in the uniaxial compaction tests, reaches a similar level of normalised dissipated energy as the tetrakaidecahedron lattice in the pillar indentation tests. They argue that for the octet lattice, W_f is a more dominant contribution to the total dissipated energy.

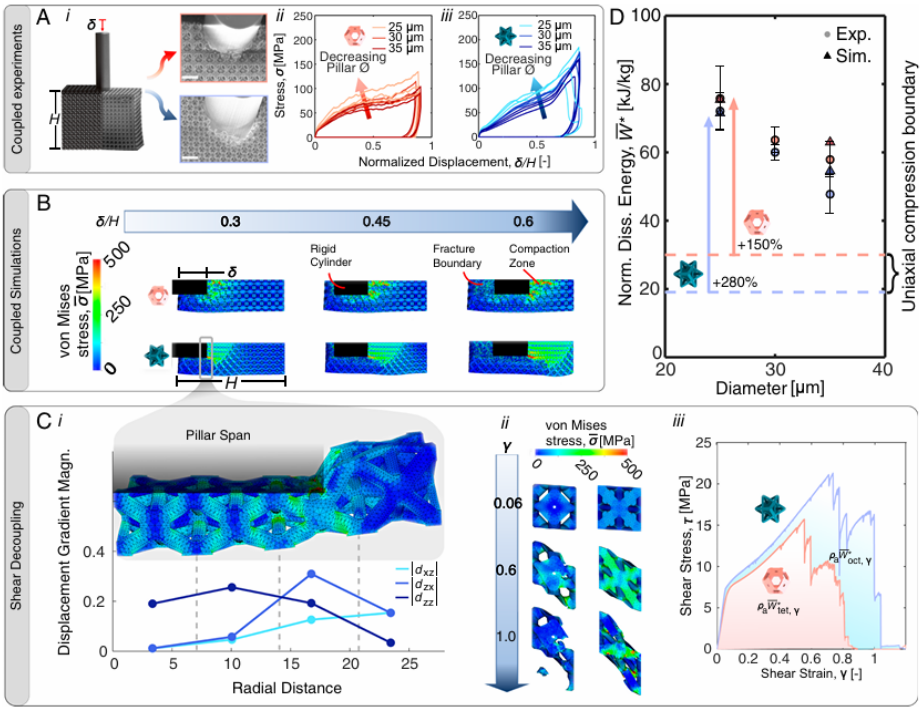


Figure 2.21: The experiments and simulations conducted on quasi-static coupled compaction-fracture behaviour. (A, i) depict the CAD model and SEM images of the experiment for an octet and a tetrakaidecahedron lattice. The scale bar indicates 10 μm. (A, ii and iii) show the stress against the normalised displacement for the two lattices for varying pillar diameters. (B) are some capture frames of the corresponding simulations for the two lattices at normalised displacements 0.3, 0.45 and 0.6. (C, i) illustrates the deformation state of the octet lattice. (C, ii) depicts a simulation of a single octet and tetrakaidecahedron unit cell subjected to pure shear. (C, iii) show the stress-strain curves of the aforementioned simulations. (D) plots the normalised dissipated energy under the pillar-indentation loading. In the same plot, the normalised dissipated energy of the compaction tests are shown as dashed lines. [12]

Butruille et al. then utilised a finite element simulation of the pillar indentation tests to find out more about the seemingly architecture dependent energy dissipation response when subjected to cylindrical fracture formation. They observed that the computed values of $\bar{W}^*$ from the simulations match the experimental data quite well for all the pillar diamaters and the architecture types, see Figure 2.21 D. The next step was determining the local deformation at fracture initiation. To this end, an approximated displacement gradient $\boldsymbol{d}$ was computed for each time step in the simulations for each unit cell (B

and C, i of Figure 2.21). The authors noticed that the deformation gradient of the unit cells near the fracture surface had comparatively large magnitude shear components, indicating a predominantly shear deformation mode.

Finite element simulations were carried out on single octet and tetrakaidecahedron unit cells subjected to simple shear with the aim of investigating how this shear-dominated deformation can affect the response, as seen in Figure 2.21 C, ii. The shear stress-strain curves for both unit cells can be seen in Figure 2.21 C, iii. The mass-normalised strain energy can then be calculated by integrating the stress-strain curves. This yields a mass-normalised dissipated energy under pure shear of 36.9 kJ/kg for the octet and 21.0 kJ/kg for the tetrakaidecahedron unit cell. The authors conclude that the fact that octet cells are more resistant to shear deformation and fracture can explain why they perform better than tetrakaidecahedron cells when it comes to particle-induced failure.

2.2.3. Bieler and Weinberg

Bieler and Weinberg study the energy absorption properties of lattice-structured materials subjected to impact loading. Since the type of lattice greatly affects the energy absorption ability of the lattice material, they study five truss-lattice structures assembled from different unit cells with three values of relative density for each configuration. All specimen are manufactured using SLA printing of a soft silicone-like material. Each specimen is then subjected to a modified Split-Hopkinson pressure bar test. Their main goal is to quantify the energy absorption performance of the different lattice configurations by calculating their specific energy absorption [6].

Experimental set-up

The five unit cells used to assemble the different specimen are summarised below with a reference to a visual.

- Octet, see Subsection 2.1.2 (Figure 2.9).
- BFCC, a face centred cubic unit cell with an extra body-centred node. For a visual of an FCC cell, see Figure 2.8.
- Diamond, a diamond lattice fit inside a cubic unit cell (Figure 2.22).
- Truncated octahedron, see Subsection 2.1.2 (Figure 2.11).
- Truncated rhombic dodecahedron (Figure 2.23).



Figure 2.22: The diamond unit cell, from [6].

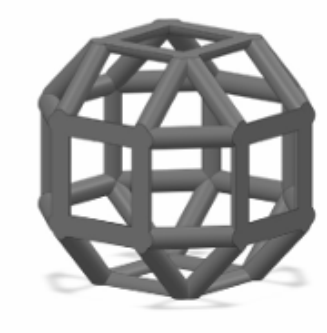


Figure 2.23: The rhombic dodecahedron unit cell, from [6].

The other geometric parameter key to the study is the relative density $\bar{\rho}$. It is equivalent to the ratio of the volume of the unit cell's members to the volume of space the unit cell occupies. See Equation 2.3.

$$\bar{\rho} = \frac{V_{\text{lattice}}}{V_{\text{ref}}} \tag{2.3}$$

V_{lattice} is the actual material volume of the unit cell, while V_{ref} is the volume of a cube with edge length L. Bieler and Weinberg use $L = 4.5$ mm for all unit cells. The aforementioned three values of $\bar{\rho}$ are $\bar{\rho} = 0.2$, $\bar{\rho} = 0.3$, $\bar{\rho} = 0.4$. Each specimen is composed of 3 x 3 x 3 unit cells.

For their material choice, Bieler and Weinberg decided to use a rather soft and ductile material to avoid destruction of the lattice under dynamic impact, and to maintain sustainability. This material is similar to thermoplastic polyurethanes (TPU), named *SuperFlex* of the company *3DMaterials*.

To carry out their impact experiments of the lattices, a modified Split-Hopkinson-Pressure bar (SHB) set-up is used. The modified SHB set-up comprises two long bars aligned along their axis. Both bars have a circular cross-section and are made of the same material. Unlike with the regular SHB set-up, the specimen is fixed to the first, incident, bar, instead of between the incident and the second, transmission, bar. A striker is accelerated to hit the specimen directly. The wave signal is measured by a strain gauge mounted on the incident bar. A schematic of this set-up can be seen in Figure 2.24.

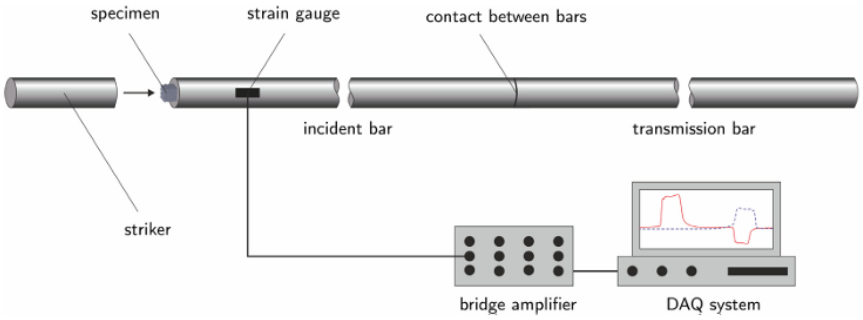


Figure 2.24: The modified SHB set-up, from [6].

For the exact details of the SHB set-up, one can consult the paper directly [6], however, the important details have been summarised below.

- Length of the bars: 3000 mm
- Length of the striker: 300 mm
- Diameter of the bars and the striker: 20 mm
- Density of the bars and the striker: 1180 kg/m^3
- Young's modulus of the bars and the striker: 3000 MPa
- Wave propagation velocity in the bars and the striker: 2500 m/s
- Impact velocity of the striker: 15 m/s

Results
The strain wave signals from the SHB tests can be integrated over time to obtain cumulative impulse over time curves using Equation 2.4.

$$p(t) = E_b \frac{\pi d_b^2}{4} \int_{t_0}^{t_1} \epsilon_I(t) \mathrm{d}t \tag{2.4}$$

Here, E_b is the elastic modulus of the incident bar, d_b is the diameter of the incident bar, and ϵ_I is the signal measured at the strain gauges of the incident bar. The cumulative impulse curves for all specimens can be found in [6], but below, the result for the octet unit cell is shown (Figure 2.25).

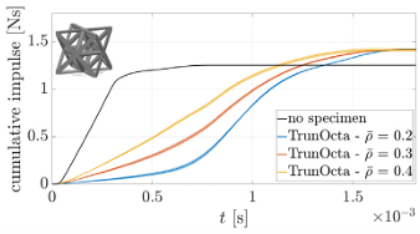


Figure 2.25: The cumulative impulse for the octet lattice for the three relative densities, from [6]

Bieler and Weinberg observe that all lattice types slow down the rise of the cumulative impulse, which is much less abrupt compared to when no specimen is present between the incident bar and the striker. Furthermore, the lower $\bar{\rho}$, the higher the slope of the cumulative impulse curve. However, the final value of the impulse for all three relative densities is mostly the same.

The next step is quantifying the energy absorption of each specimen. For a standard SHB test, this is calculated from the difference of the work done by the incident wave, W_I, and the sum of the work done by the reflected and transmitted pulses, W_R and W_T, see Equation 2.5.

$$\Delta W = W_I - (W_R + W_T) \tag{2.5}$$

ΔW is the work absorbed by the specimen. To obtain W_I, W_R and W_T, one must integrate the corresponding strain pulses. Since Bieler and Weinberg used a modified SHB set-up, they only need to find W_I. This is calculated using Equation 2.6.

$$W_I = A_b c_b E_b \int_0^t \epsilon_I^2(\bar{t}) \mathrm{d}\bar{t} \tag{2.6}$$

Here, A_b is the cross-sectional area of the incident bar. c_b is the wave speed in the incident bar. Because they only consider the work done by the incident pulse, the absorbed energy for each specimen is determined from the difference of a reference measurement, W_{ref}, without specimen, and a measurement with the specimen, W_{lattice}. In equation form:

$$\Delta W = W_I^{\text{ref}} - W_I^{\text{lattice}} \tag{2.7}$$

The reference energy has the value $W_I^{\text{ref}} = 11.37$ Nm for striker velocity of 15 m/s.

The total absorbed energies (EA) for each specimen can be seen in Figure 2.26.

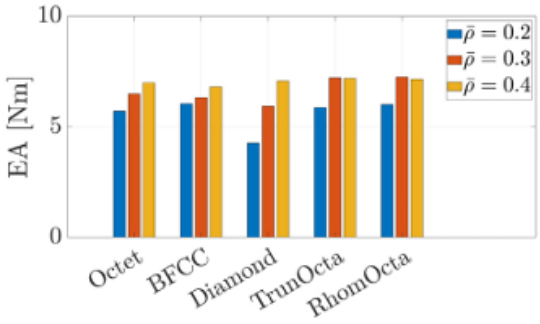


Figure 2.26: The absorbed energy for each lattice of each relative density, from [6]

One can see that for octet, BFCC and diamond lattices, the specimens with relative density $\bar{\rho} = 0.4$ have the highest energy absorption. Meanwhile, for the truncated octahedron and rhombic dodecahedron lattices, the specimens with relative density $\bar{\rho} = 0.3$ and $\bar{\rho} = 0.4$ have very similar energy absorption.

However, it is more informative to look at the specific energy absorption (SEA) to compare the energy absorption properties of each lattice structure. The specific energy absorption is calculated using Equation 2.8.

$$\text{SEA} = \frac{\Delta W}{m} \tag{2.8}$$

Here, m is the mass of the specimen. It is calculated using Equation 2.9.

$$m = 27\rho\bar{\rho}L^3 \tag{2.9}$$

Above, 27 is the number of unit cells and ρ is the mass density of the resin. The specific absorbed energy of each specimen is shown in Figure 2.27.

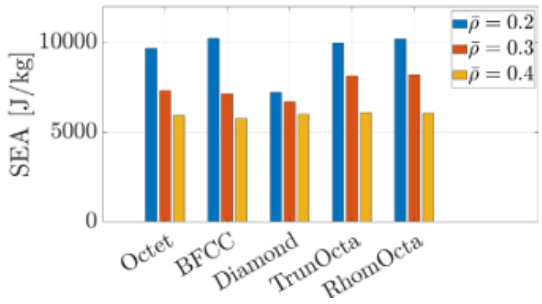


Figure 2.27: The specific absorbed energy for each lattice at each relative density, from [6]

Bieler and Weinberg observe that lower $\bar{\rho}$ results in higher SEA across all five unit cell configurations. Increasing $\bar{\rho}$ decreases SEA for all lattice types except for the diamond, where SEA is not greatly affected by $\bar{\rho}$.

Looking at both the total absorbed energy and the specific absorbed energy, they contend that the truncated octahedron and rhombic dodecahedron structures have the best energy absorption. They also note that both types of lattice structures have high negative Maxwell numbers, which is indicative of bending-dominated structures, and feature no nodes in the centre or corners of the unit cells.

2.2.4. Summary of impact experiments of truss-lattice metamaterials review

Portela et al. (2021) show experimentally that nanoarchitected materials boast far superior specific energy absorption compared to standard ballistic-grade materials like Kevlar [64]. They conclude that additional dissipation mechanisms which boost the impact efficiency of architected materials are the

result of the material's ability to propagate compaction shocks.

Building on the results from [64], Butruille et al. (2024) investigated the influence of architecture on impact performance, and in addition, they made an effort to decouple the dissipation mechanisms to better understand how the specific architecture of a cell can affect total dissipation [12]. They demonstrate that an octet lattice outperforms a tetrakaidecahedron lattice in terms of specific energy dissipation. In their effort to decouple the dissipation mechanisms, they indicate that dynamic compaction of the material and fracture caused by impact are significant components of the total dissipated energy. They discovered that an octet lattice performed worse than a tetrakaidecahedron lattice when subjected to quasi-static uniaxial compaction, despite its superior energy dissipation in the particle impact tests. They concluded that this hints at architecture-dependent fracture playing a key role in the energy dissipation performance of architected materials.

Bieler and Weinberg (2025) conclude that the truncated octahedron lattice had the best energy absorption performance in their experiments [6]. Since a truncated octahedron is a tetrakaidecahedron, it may seem like a contradiction of [12], but the experiment performed by Bieler and Weinberg is a form of uniaxial compaction, which, as Butruille et al. (2024) have discovered, gives tetrakaidecahedron lattices the edge in energy absorption performance over octet lattices [12].

2.3. Impact physics

In order to model impact problems and interpret the impact response of the specimen, it is necessary to understand the physics involved in an impact problem. For this purpose, this section covers impact problems in different velocity ranges, and the associated physical phenomena in the target material. In particular, the focus is on ballistic impact problems, which is especially relevant for aerospace structures.

2.3.1. Impact regimes

The ballistics community generally define impact regimes based on the impact velocity ranges [65]. One study divides ballistic impact problems into ordnance ranges. Here, projectile velocities ranging from 500 to 1300 m/s fall in the nominal ordnance range. Impact problems with a velocity range of 25 to 500 m/s are categorised as subordnance. Impact problems slower than this are mostly limited to drop tests and other experiments. On the other side of the ordnance domain, impact velocities exceeding 1300 m/s up to 3000 m/s belong to the ultraordnance domain. When the impact velocity goes beyond 3000 m/s, is when the hypervelocity range begins [66].

However, assigning hard velocity boundaries on each impact regime implicitly assumes that the physical phenomena that occur in the material during impact are directly correlated with the impact velocity [65]. Instead, for the purposes of covering the impact phenomena, hard velocity boundaries are omitted, and the physical processes are only divided in two groups: low velocity impact and high velocity impact. The reason for this divide is the significant change in material behaviour when the impact velocity becomes high enough [67] [65].

2.3.2. Low velocity impact

The impact phenomena in this group are material strength dominated [65]. The mechanisms encountered here are usually tied to elastic and/or plastic waves [66]. If the velocities are high enough, impact causes localised plastic deformations [65] and possibly fracture. Numerical models intending to capture plastic or fracture behaviour must be equipped with suitable constitutive and cohesive models.

Elastic deformations

Elastic deformations in impact problems are generally only observed at very low impactor velocities, usually in laboratory settings. The impact velocity must not cause the contact pressure to exceed the compressive yield strength of the specimen, or the projectile [66]. Elastic deformations are not very interesting processes when investigating energy dissipation mechanisms in materials for impact-protection since no energy dissipating phenomena occur during impact.

Plastic deformations
If the impact velocity does result in contact pressures which exceed the material's yield strength, then plastic deformations occur. The dissipation of impact energy can be described with elastic-plastic mechanical and constitutive behaviour such as strain hardening and permanent deformation. When the impact velocity stays below the stress wave velocity in the material, then the plastic waves are transmitted through a volume far larger than the dimensions of the projectile, but they rapidly transition into elastic waves. When impact velocities become higher, this affected volume becomes smaller and smaller. If the pressures caused by the impact exceed the material's flow stress, then hydrodynamic effects start becoming more important than material strength effects [65], see Subsection 2.3.3.

Plasticity in truss-lattice metamaterials means the struts undergo plastic deformations. In principle, the abovementioned mechanisms still hold since the struts can exhibit strain hardening and permanent deformations.

Fracture
In thin or moderately thick bulk materials, fracture due to impact usually causes perforation. The processes that go into perforation differ and depend on material properties, geometry and the impact velocity. According to Backman and Goldsmith (1978), the most common mechanisms are fracture due to the initial compression wave, radial fracture, spalling, scabbing, plugging, front or real petalling, fragmentation and ductile hole enlargement [66] (Figure 2.28).

Figure 2.28 illustrates how the fracture mechanisms depend on material properties and geometry. For instance, radial fracture, plugging and fragmentation are often encountered in brittle materials such as ceramics or thin, hard metal plates. Meanwhile, scabbing and petalling are ductile failure phenoma characterised by large deformations and/or plastic flows [66]. Note that not all these fracture processes occur at low velocities.

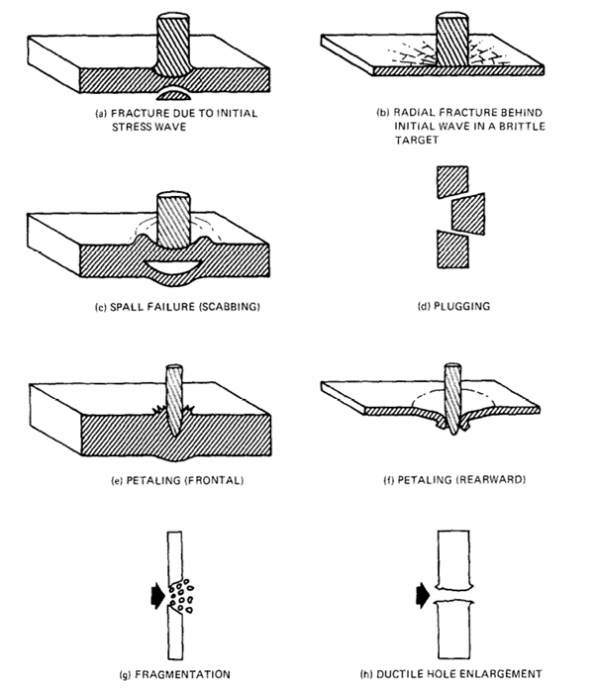


Figure 2.28: Perforation mechanisms, from [66].

Initiating the fracture surfaces and propagating the crack costs energy, which means that material fracture is also a way of disispating impact energy.

In the case of truss-lattice metamaterials, the fracture modes will not be as diverse as those just mentioned for bulk materials. Instead, the fracture modes are limited to the fracture modes encountered in struts: axial, transverse, bending and torsion [16]. This is treated further in Section 2.7. Still, fracture plays a similar role in energy dissipation in that it still costs energy to initiate and propagate cracks.

2.3.3. High velocity impact

The impact phenomena in this group are tied to fluid-like behaviour. This is where hydrodynamics starts taking centre stage, becoming more dominant than material strength effects. With hydrodynamics come shock wave initiation and propagation. In addition, thermal effects become important as well [65]. Numerical models aimed at simulating high velocity impact must therefore take hydrodynamic and thermal effects into account.

Hydrodynamics

In high velocity impact the initiation and propagation of shock waves become dominant phenomena. These shock waves are created from the compressive waves caused by the impact. When the shock waves reach a material interface or a free surface, tensile waves are created to satisfy the boundary conditions [68]. These rarefraction waves may cause a pressure that exceeds the tensile strength of the material. This results in spalling, which is technically a way of dissipating impact energy [65], but for impact-protection applications, it is hardly ideal that the protective material ejects high speed fragments the other way.

The shock waves also cause discontinuities in for instance material density and temperature, which necessitates consideration for thermodynamic effects. This puts even more emphasis on the equation of state for describing material behaviour in high velocity impact. When the material is first struck, the entropy of the system rises. The expansion back to the original state of zero pressure is approximately isentropic, which means the material retains a higher level of entropy. This is indicative of changes in the microstructure of the material since shocks can increase the temperature of parts of the material. Stronger shocks can even result in severe thermal effects like melting or vaporisation [65].

Both the effects of shock waves on the fracture and temperature behaviour of the material are strong reasons why numerical simulations of high velocity impact necessitate a hydrodynamic model. In the case of truss-lattice metamaterials, there is a rather unique opportunity to design the topology of the lattice in such a way so as to manipulate the shock waves since they can only travel through the material, which is the struts. This could be a method for architected materials to leverage the dissipation mechanisms of shock waves. However, the extreme localised heating associated with the same shock waves at very high velocities can still be troublesome if they cause the struts to melt.

Plastic heating

For high velocity impact, thermal effects are closely tied to mechanical behaviour. For instance, most of the work done during plastic deformation is turned into heat. The local temperature increase as a result of plastic work done is given by Equation 2.10 [65].

$$\Delta T = \beta \frac{\Delta W_{pl}}{\rho c_v} \tag{2.10}$$

Here, ΔT is the temperature change, β is the Taylor-Quinney (TQ) coefficient, ΔW_{pl} is the plastic work done, ρ is the mass density of the material, and c_v is the specific heat capacity of the material at constant volume. β is an experimentally determined value. When Equation 2.10 is combined with a temperature dependent constitutive law, then the thermomechanical dynamic response of a material subjected to plastic deformation can be modelled [65].

Shock heating

At very high impact velocities, the generated shock waves travel at a higher speed than the bulk speed of sound in the material. The kinetic energy of the projectile which is divided over the material and the impactor at impact, is thus also partially turned into internal energy, increasing the local temperature of the material. The local increase in internal energy can lead to melting or vaporisation around the area affected by the impactor. For thin materials, the shock and refraction can shatter the material, resulting

in debris clouds of molten ejecta [65].

The effect of shock heating is described by Equation 2.11 [65].

$$dT = \frac{1}{c_v}\left[dE - \left(\frac{\partial E}{\partial V}\right)_T dV\right] \tag{2.11}$$

2.3.4. Energy absorption mechanisms

The energy dissipation mechanisms involved with impact mentioned earlier are summarised below. They have been consolidated into one list for convenience so that one has a list ready when interpreting the results from the impact simulations. Each item will refer to the part of this section it was mentioned in. The two last entries are not dissipation modes and were not mentioned before in this section. However, they do contribute to the energy absorption in impact problems, so they have been included in this list.

- **Plastic deformation:** Energy is dissipated to cause permanent deformations (Subsection 2.3.2).
- **Fracture:** Energy is dissipated through the initiation and growth of fracture surfaces (Subsection 2.3.2).
- **Shock waves:** Energy is dissipated by localised heating and deformations (Subsection 2.3.3).
- **Melting and vaporisation:** Enough energy is turned into heat to cause melting or vaporisation (Subsection 2.3.3).
- **Kinetic energy transfer:** The inertial contributions to the total absorbed energy of a material resulting from imparting kinetic energy onto the ejected fragments of the material (Subsection 2.2.2).
- **Densification:** Densification or compaction is another contribution to the total absorbed energy. It is the result of a lattice material being compressed to such a degree that it starts behaving like bulk material. This is due to the struts being unable to deflect further because of contact with neighbouring struts [15]. This bulk behaviour causes a stiffening effect also seen in D and E of Figure 2.20. To capture this behaviour, a numerical model must be able to capture contact between the struts of the lattice metamaterial.

2.3.5. Summary of impact physics

As this section has shown, an impact problem involves many different processes, and which processes are present depends on the impact velocity, material properties and geometry. When modelling impact problems, these different phenomena will need to be accounted for, which informs the approach taken to develop a computational model. For low velocity impact, one mainly is concerned with elastic and plastic deformations, and possible brittle fracture such as radial fracture, or ductile fracture like petalling caused by those deformations. For high velocity impact, shock waves and thermal effects become more dominant than material strength behaviour.

2.4. Computational modelling approaches for truss-metamaterial impact problems

The prior work on impact problems involving truss metamaterials consulted in this thesis all underline how crucial it is to the design of said materials to better understand the energy-absorbing capabilities of cellular materials in the dynamic regime [64] [12] [6]. The logistics of performing high-speed impact tests on representative specimens are not trivial, so being able to simulate dynamic impact performance goes a long way in the design process of metamaterials designed for impact.

To inform the methodology used in this thesis for modelling truss-metamaterial impact problems, a number of computational studies of impact involving (truss-based) architected materials were examined. Among these are the aforementioned works of Butruille et al. (2024) [12], and Bieler and Weinberg (2025) [6]. The other works which were consulted are the following. Ayers et al. (2013) study auxetic metamaterials subjected to concentrated impact loads [10]. Gaitanaros and Kyriakides (2014) simulate dynamic crushing of open-cell aluminium foams using micromechanically accurate foam models [13]. Mueller et al. (2019) conducted a computational study of quasi-static and dynamic energy absorption

performance of periodic and stochastic 3D lattice materials [14]. Dattelbaum et al. (2020) studied the shock wave mitigation performance of fractal structures with controlled arrangements of voids [11]. Crone et al. (2022) present a multi-scale model to capture the response of nanoarchitected materials subjected to large deformations [15].

In the following, the modelling approaches taken in the computational studies mentioned above are treated. To help inform what a computational model for impact problems involving truss metamaterials must comprise of, the state-of-the-art of modelling impact in architected materials is described using three specific facets of each paper's numerical framework: the choice of finite element type, modelling inter-element contact and representing damage.

2.4.1. Choice of elements

The first facet of the state-of-the-art computational models to be examined is the choice of finite element. This forms the basis of the behaviour of the bulk material, however, it also affects the choices made regarding modelling contact and damage because of the differences between element types.

The literature generally uses either of two approaches to model the material: continuum elements ([10] [11] [12] [6]) and beam elements ([13] [14] [15]). Continuum elements are 3D solid elements which maintain the full 3D stress and strain states and make no kinematic assumptions. However, they require lots of degrees of freedom per element, which means that upscaling a continuum element lattice model increases computational cost quite quickly. By contrast, beam elements are 1D structural elements. They are very efficient when used correctly and as such, upscaling a lattice based on beam elements is not as costly. That said, using beam elements means making geometric assumptions on e.g. the cross-section or initial curvature, or making assumptions on deformation modes. This means beam elements have a limited range of validity, depending on the beam formulation.

2.4.2. Modelling inter-element contact

The next component of state-of-the-art models is whether contact between elements is modelled. Densification is a key phenomenon in architected materials which are rapidly compressed [15], so it would make sense that the state-of-the-art numerical frameworks model self-contact to capture this effect.

Most authors explicitly state they model self-contact between the metamaterial members during their simulations. Gaitanaros and Kyriakides (2013), and Mueller et al. (2019) go one step further and specify they used a penalty-based contact enforcement technique. Only Ayers et al. (2013) and Dattelbaum et al. (2020) do not explicitly mention whether they model self-contact.

2.4.3. Representing damage

The final aspect of the state-of-the-art for modelling impact in metamaterials is the representation of damage in the metamaterials. In Section 2.3, plasticity and fracture are mentioned as important dissipation mechanisms for impact. As such, modelling impact problems involving metamaterials must involve representing damage in the struts as a result of the loading.

If plasticity is viewed as a form of damage, then most authors account for damage in their computational models. Only Bieler and Weinberg (2025) do not represent either plasticity or fracture because their material is very soft, and during their experiments they showed that the material would assume its original state after compaction, so they did not have a need to model plasticity or failure. The other authors used elastic-plastic constitutive models to capture plastic behaviour, except for Dattelbaum et al. (2020) who used a visco-plastic material model.

However, only Ayers et al. (2013), Mueller et al. (2019) and Butruille et al. (2024) explicitly mention they model material fracture. All three works utilise a form of damage evolution model to erode and eventually eliminate a failed element.

2.4.4. Conclusion on the state-of-the-art of metamaterial impact modelling

After examining the approaches to modelling impact problems in metamaterials from the literature, one can identify the main components of the state-of-the-art computational framework for modelling impact in architected materials. These components have been summarised below.

- The struts are either modelled using continuum elements or beam elements.
- Contact between struts is modelled.
- Damage in the struts is modelled using plasticity and sometimes ductile material failure.

Notably, the literature does not model material failure through crack initiation and growth, but rather through damage evolution and element erosion. In fact, there seems to be lack of a state-of-the-art simulation framework for the dynamic impact of truss-based metamaterials which unifies self-contacting and fracture behaviour of beam-based struts. Works like Kota et al. (2024) [16] formulate a beam-fracture framework, but a model coupling contact and fracture in beams is yet to be developed. This serves as one of the motivating factors behind the formulation of the unified contact and fracture finite element framework for beam-lattice metamaterials.

A unified contact and fracture framework

The list from earlier can be turned into a blueprint for a computational framework that unifies inter-element contact with fracture. When applied to beam-lattice metamaterials, the blueprint reads:

1. The struts are modelled with beam elements.
2. Contact between struts is governed by a beam-to-beam contact model.
3. Strut failure is captured with a brittle fracture model.

Compared to the list from before, some modelling choices have been made. The framework uses beam elements to model the struts because of their computational efficiency. Given that the framework is applied to beam-lattice metamaterials, this seems like a fitting choice, especially when the relative density of the lattice is low enough to result in slender struts. The beam formulations considered are the topic of Section 2.5.

Representing the struts with beam elements naturally means the contact model is a beam-to-beam contact model. Beam-to-beam contact models can leverage the fact that beam elements are 1D elements, which also benefits computational efficiency. The contact formulation is covered in Section 2.6.

Finally, damage and failure of the struts is described by a fracture model that uses crack initiation and growth rather than damage evolution. The choice of a fracture model over a damage evolution model is a conscious decision so that the computational framework fills a niche which is not really represented in the literature: one that unifies fracture in beams with contact between beams. The fracture model is further treated in Section 2.7. Although it is possible to represent plastic behaviour through the constitutive model of the beam elements, the computational framework synthesised in this thesis does not consider plasticity for simplicity.

2.5. Overview of beam theories

The first component of the blueprint mentioned in Subsection 2.4.4 is modelling the struts with beam elements. Beam elements are particularly popular choices for modelling slender bodies. Historically, the motivation for using beam theories was the availabiltiy of analytical solutions, but today their popularity stems from their higher computational efficiency and their numerical well-posedness when applied to slender structures compared to 3D continuum theories [69].

Beam theories can generally be divided into three fundamental groups: induced beam theories, intrinsic beam theories and semi-induced beam theories. Induced beam theories are essentially reduced 1D continuum theories which are derived from 3D continuum mechanics. With kinematic constraints, as well as kinetic and constitutive resultants, induced beam theories are able to model the motion and deformation of slender continua in 3D space using a 1D body. For induced beam theories, the aforementioned resultants are derived through integration over the beam cross-section. Intrinsic beam

theories directly postulate the 1D resultant quantities without deriving them from 3D continuum mechanics. The resultants are still mechanically consistent but are simply decoupled from 3D continuum theory. Usually, the postulated constitutive relations are determined experimentally. The last group, semi-induced beam theories, are the middleground between induced and intrinsic theories. These theories consistently derive the kinematic and kinetic relations from the 3D continuum theory, but postulate the constitutive relations [69].

There is a large body of work on beam theories, meaning there is a large choice of different beam elements. Each beam formulation has its particularities, so to inform the eventual choice of beam element for the framework, this section will go over the formulations of a select number of beam theories. To limit the scope of this literature study, the beam theories considered are the ones mentioned in an overview by Meier et al. (2017) [69].

2.5.1. Euler-Bernoulli theory

The first beam formulation to be covered is the most classical beam theory: the Euler-Bernoulli beam. This beam model is the simplest, and it is a staple of nearly every structural mechanics class. Daniel Bernoulli, one of the contributors to what is now known as the Euler-Bernoulli beam theory, proposed an engineering solution to the cantilever bending problem in the 18th century [70]. By assuming small deformations, $ds \approx dx$, his formulation for the cantilever problem reads as:

$$EI\frac{d^2y}{dx^2} = Wx \tag{2.12}$$

This equation has an analytical solution for y, the lateral displacement as a result of the lateral load W, obtained through simple integration. The simplicity of the Euler-Bernoulli model comes at a cost. Aside from assuming small deformations, resulting in a linear theory, the formulation is restricted to initially straight beams. Lastly, Bernoulli's hypothesis forms the foundation of Euler-Bernoulli beams: beam cross-sections perpendicular to the beam centreline before deformations remain perpendicular after deformations, and beam cross-sections remain plane and unwarped when deformed [71]. The first part of the Bernoulli hypothesis is the shear-rigid assumption. This is also illustrated in Figure 2.29.

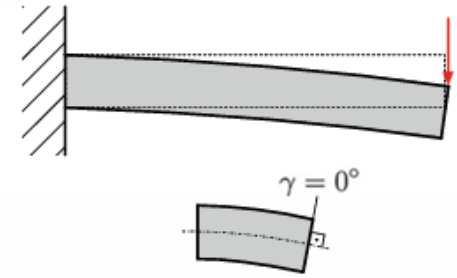


Figure 2.29: The deformation mode of a bending beam under the shear-rigid assumption from [71].

In an Euler-Bernoulli beam, only normal stress, which is linearly distributed across the cross-section (in the case of pure bending), causes deformation. This causes maximum tension on one side of the cross-section (the top face in Figure 2.29), and maximum compression on the other side (the bottom face in Figure 2.29). If the cross-section is symmetric, then there is a neutral axis which bisects the cross-section [71].

Equation 2.12 is only formulated for transverse forces, but it is possible to account for axial and torsional deformations by formulating similar governing forces for the axial and torsional degree of freedoms, keeping in mind that Equation 2.12 is essentially only for transverse deflections.

To summarise, the Euler-Bernoulli beam formulation is a simple, but effective model for describing initially straight beams subjected to different loading conditions. By reducing a 3D problem to a 1D problem governed by a scalar differential equation, the Euler-Bernoulli model allows for straightforward

analytical analysis. That said, the assumption of small deformations and the shear-rigid assumption mean that Euler-Bernoulli beams are generally only applicable to long, slender beams subjected to small deflections.

2.5.2. Timoshenko theory

The main difference between the Timoshenko beam and the Euler-Bernoulli beam is the fact that the Timoshenko beam is shear-flexible, meaning that effect of shear on the deformations is captured in the model. Normally, the Bernoulli hypothesis requires beam cross-sections to remain orthogonal to the neutral line when deformed, however, the Timoshenko formulation allows beam cross-sections to break orthogonality with the neutral line through a shear deformation. Timoshenko theory approximates the shear stress distribution through the thickness as constant, but a real shear stress distribution is variable. Therefore, a Timoshenko beam requires a shear correction factor to ensure the constant shear distribution is equivalent to the real shear distribution when integrated through the thickness [71]. The deformation mode of the Timoshenko beam in the xy-plane can be seen in Figure 2.30.

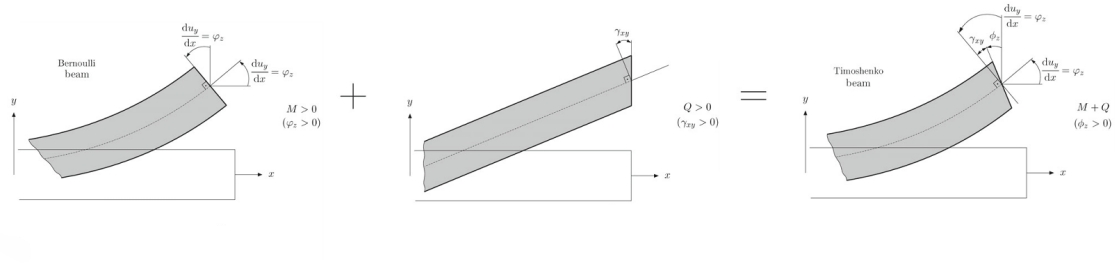


Figure 2.30: The superposition of the Euler-Bernoulli beam and the shear deformation, resulting in a Timoshenko beam [71].

The governing equation for the lateral deflection of a Timoshenko beam is given by Equation 2.13 [71].

$$\frac{dM_z(x)}{dx} = \frac{d}{dx}\left(EI_z \frac{d\phi_z(x)}{dx}\right) \tag{2.13}$$

This differential equation is not that different from the governing equation for Euler-Bernoulli beams. In fact, if $\phi_z = \frac{dy}{dx}$ holds, which is to say, if there are no shear deformations, then one gets the governing equation for the lateral deflections of an Euler-Bernoulli beam.

The fact that Timoshenko beams can model the effect of shear deformation on the beam response presents a huge advantage over the Euler-Bernoulli beam. This enables Timoshenko beams to remain valid for less slender beams compared to Euler-Bernoulli beams [71] [72]. One caveat is that Timoshenko beam theory introduces an extra variable because of the shear effect, increasing computational cost compared to the Euler-Bernoulli model. Also, because the Timoshenko formulation is essentially an Euler-Bernoulli model that can capture shear deformations, its range of application is still limited to initially straight beams subjected to small deformations.

2.5.3. Simo-Reissner theory

The overview of Simo-Reissner beam theory and the formulae covered here were presented in study by Meier et al. (2017) comparing geometrically exact finite element formulations for slender beams [69].

A small history lesson is in order to introduce the Simo-Reissner beam formulation. It starts with Kirchhoff who proposed a new beam theory in 1859, which significantly improved upon the Euler-Bernoulli formulation. This model marks the first time a beam formulation could describe beams with arbitrary initial curvatures and large 3D deformations, which included bending and torsion modes [73]. In order to capture small axial tension, a modification to this formulation was made by Love in 1944 [74].

Following these works, Reissner later rounded out the beam formulation with two additional deformation measures aimed at capturing shear deformation for the 2D case [75] and the 3D case [76]. Reissner's formulation for the 3D case included some additional approximations, but Simo improved upon Reissner's work, resulting in a formulation that is more consistent with 3D continuum theory, with only the constitutive law being postulated [77]. This semi-induced beam theory used to be called geometrically exact beam theory, but it is now also known as Simo-Reissner beam theory [69].

Before moving on to the mathematical formulation, it is worthwhile to explain the term *geometrically exact beam*. According to Simo, a beam formulation can be seen as geometrically exact when the following holds: "the relationships between the configuration and the strain measures are consistent with the virtual work principle and the equilibrium equations at a deformed state *regardless of the magnitude of displacements, rotations and strains*" (Crisfield et al. (1999), p. 1126) [78]. This is also why Simo used the notion of finite strain in his work [77], since no simplifying assumptions are made on the magnitude of the deformations and rotations.

Simo-Reissner beam theory requires six degrees of freedom to uniquely define the position of the centroid and the orientation of the cross-sections because of the assumption of rigid cross-sections. As a result, a Simo-Reissner beam can be seen as a 1D Cosserat continuum which is derived from a 3D Boltzmann continuum [69]. With this, the mathematical formulation of the Simo-Reissner beam can be treated. The formulation has been summarised by Meier et al. (2017) [69], and were originally posed by Reissner [75] [76], and Simo and Vu-Quoc [77] [79].

Kinematics
The first stop is the kinematics of the Simo-Reissner beam. Here, only the most relevant features of the kinematics are treated. Figure 2.31 shows the main kinematic quantities for the Simo-Reissner beam.

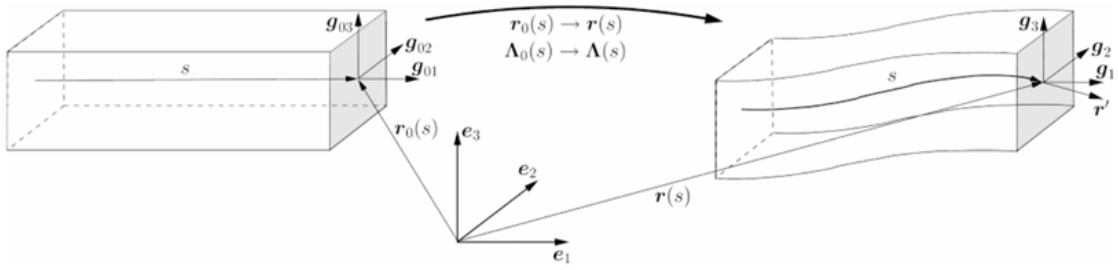


Figure 2.31: Kinematic quantities defining the initial and deformed configuration of the Simo-Reissner beam, from [69].

Starting in the undeformed configuration, the centreline of the beam is represented by the space curve $s \mapsto \boldsymbol{r}_0(s) \in \mathbb{R}^3$. All centroids of the cross-sections lie on this curve. $s \in [0, l] := \Omega_l \subset \mathbb{R}$ is the arc-length parameter of the curve with $l \in \mathbb{R}$ the undeformed length of the beam. The undeformed state is also defined by a set of right-handed orthonormal triads. This field is defined as $s \mapsto \boldsymbol{g}_{01}(s), \boldsymbol{g}_{02}(s), \boldsymbol{g}_{03}(s) \in \mathbb{R}^3$, and the triads are also known as material traids. The material triads are then fixed to the beam cross-sections, which are undeformable following the assumption of rigid cross-sections. The triad $\boldsymbol{g}_{01}$ seen in Figure 2.31 is defined as $\boldsymbol{g}_{01}(s) = \boldsymbol{r}_0'(s)$, being the unit tangent vector to the initial centreline. From this, $\boldsymbol{g}_{02}(s)$ and $\boldsymbol{g}_{03}(s)$ point in the direction of the principal axes of inertia of the cross-section at s [69].

The rotation from the global frame $\boldsymbol{E}_i$ to the initial local frame $\boldsymbol{g}_{0i}(s)$ is defined by the orthogonal transformation $s \mapsto \boldsymbol{\Lambda}_0(s) \in SO(3)$. This results in $s \mapsto C_0 := (\boldsymbol{r}_0(s), \boldsymbol{\Lambda}_0(\boldsymbol{\Psi}_0(s))) \in \mathbb{R}^3 \times SO(3)$ for the undeformed configuration. It follows that the deformed configuration at time $t \in \mathbb{R}$ is defined by $s, t \mapsto C := (\boldsymbol{r}(s,t), \boldsymbol{\Lambda}(\boldsymbol{\Psi}(s,t))) \in \mathbb{R}^3 \times SO(3)$. Here, the orthogonal transformation $\boldsymbol{\Lambda}(\boldsymbol{\Psi}(s,t))$ maps from the global frame to the current local frame $\boldsymbol{g}_i(s)$, and $\boldsymbol{\Psi}$ is the rotation vector. Because of shear deformations, the base vector $\boldsymbol{g}_1(s,t) := \boldsymbol{g}_2(s,t) \times \boldsymbol{g}_3(s,t)$ generally need not be tangent to the deformed centreline of the beam.

This concludes the brief overview of the kinematics of Simo-Reissner theory. The next part of the theory to be covered is the formulation of the weak form and the derivation of the deformation measures.

Weak form and deformation measures
The first step towards formulating the weak form is defining the strong form. As such, the strong form of the equilibrium equations is given by Equation 2.14 [69].

$$\boldsymbol{f}' + \tilde{\boldsymbol{f}} + \boldsymbol{f}_\rho = \boldsymbol{0} \tag{2.14a}$$
$$\boldsymbol{m}' + \boldsymbol{r}' \times \boldsymbol{f} + \tilde{\boldsymbol{m}} + \boldsymbol{m}_\rho = \boldsymbol{0} \tag{2.14b}$$

Here, $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{m}}$ are distributed external forces and moments per unit length, respectively. $\boldsymbol{f}_\rho$ and $\boldsymbol{m}_\rho$ are the inertial forces and moments. Finally, $\boldsymbol{f}$ and $\boldsymbol{m}$ represent the force and moment stress resultants acting on the cross-sectional area of the beam.

Inserting the material stress resultants $\boldsymbol{F} := \boldsymbol{\Lambda}^T \boldsymbol{f}$ and $\boldsymbol{M} := \boldsymbol{\Lambda}^T \boldsymbol{m}$ into Equation 2.14 yields a material form of the 1D equilibrium equations.

Following standard variational techniques, the variations $\delta\boldsymbol{r}(s) \in \mathbb{R}^3$ and $\delta\boldsymbol{\theta}(s) \in \mathbb{R}^3$ are defined as virtual displacements and virtual rotations, respectively. When the principle of virtual work is used on Equation 2.14 with the two admissible variations $\delta\boldsymbol{r}$ and $\delta\boldsymbol{\theta}$, followed by performing integration by parts, one obtains the spatial weak form Equation 2.15 [69].

$$\begin{aligned} &\int_0^l \left[\delta\boldsymbol{\theta}'^T \boldsymbol{m} + (\delta\boldsymbol{r}' - \delta\boldsymbol{\theta} \times \boldsymbol{r}')^T \boldsymbol{f} - \delta\boldsymbol{\theta}^T(\tilde{\boldsymbol{m}} + \boldsymbol{m}_\rho) - \delta\boldsymbol{r}^T(\tilde{\boldsymbol{f}} + \boldsymbol{f}_\rho)\right] \mathrm{d}s \\ &- \left[\delta\boldsymbol{r}^T \bar{\boldsymbol{f}}\right]_{\Gamma_\sigma} - \left[\delta\boldsymbol{\theta}^T \bar{\boldsymbol{m}}\right]_{\Gamma_\sigma} \doteq 0 \end{aligned} \tag{2.15}$$

Above, $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{m}}$ are the external forces and moments at the Neumann boundary Γ_σ of the beam. Meier et al. use $\boldsymbol{f}_\sigma$ and $\boldsymbol{m}_\sigma$, but to keep consistency with Subsection 2.5.4, this notation was used.

To obtain the objective spatial deformations, the orthogonal transformation $\boldsymbol{\Lambda}$ can be used [69], resulting in Equation 2.16.

$$\boldsymbol{\gamma} = \boldsymbol{r}' - \boldsymbol{g}_1 \tag{2.16a}$$
$$\boldsymbol{\omega} = \boldsymbol{k} - \boldsymbol{\Lambda}\boldsymbol{\Lambda}_0^T \boldsymbol{k}_0 \tag{2.16b}$$

Here, $\boldsymbol{k}$ is the spatial curvature vector and $\boldsymbol{k}_0$ is the initial curvature vector.

If Equation 2.14 was used in its material form, then the objective material deformation measures $\boldsymbol{\Gamma}$ and $\boldsymbol{\Omega}$, which are work-conjugates of $\boldsymbol{F}$ and $\boldsymbol{M}$, respectively, are given by Equation 2.17.

$$\boldsymbol{\Gamma} = \boldsymbol{\Lambda}^T \boldsymbol{r}' - \boldsymbol{E}_1 \tag{2.17a}$$
$$\boldsymbol{\Omega} = \boldsymbol{K} - \boldsymbol{K}_0 \tag{2.17b}$$

Using the objective deformations from Equation 2.16 and Equation 2.17, the objective variations are given by Equation 2.18 and Equation 2.19 [69].

$$\delta_0 \boldsymbol{\gamma} = \delta\boldsymbol{r}' - \delta\boldsymbol{\theta} \times \boldsymbol{r}' \tag{2.18a}$$
$$\delta_0 \boldsymbol{\omega} = \delta\boldsymbol{\theta}' \tag{2.18b}$$

$$\delta\boldsymbol{\Gamma} = \boldsymbol{\Lambda}^T (\delta\boldsymbol{r}' + \boldsymbol{r}' \times \delta\boldsymbol{\theta}) \tag{2.19a}$$
$$\delta\boldsymbol{\Omega} = \boldsymbol{\Lambda}^T \delta\boldsymbol{\theta}' \tag{2.19b}$$

In Equation 2.19, the components of $\boldsymbol{\Gamma}$ represent axial tension and shear, and the components of $\boldsymbol{\Omega}$ represent bending and torsion.

Constitutive relations
The last component of Simo-Reissner beam theory which must be covered, are the constitutive relations between the stress resultants $\boldsymbol{F}$ and $\boldsymbol{M}$, and the deformation measures $\boldsymbol{\Gamma}$ and $\boldsymbol{\Omega}$. For simplicity, a hyperelastic constitutive law is given by Equation 2.20 [69].

$$\tilde{\Pi}_{int}(\boldsymbol{\Gamma}, \boldsymbol{\Omega}) = \frac{1}{2}\boldsymbol{\Gamma}^T \boldsymbol{C}_F \boldsymbol{\Gamma} + \frac{1}{2}\boldsymbol{\Omega}^T \boldsymbol{C}_M \boldsymbol{\Omega} \tag{2.20}$$

Here, $\tilde{\Pi}_{int}$ is the length-specific hyperelastic internal energy function. The explicit constitutive relation between $\boldsymbol{F}$ and $\boldsymbol{\Gamma}$, and $\boldsymbol{M}$ and $\boldsymbol{\Omega}$ can be obtained by taking the partial derivative of Equation 2.20 with respect to the related deformation measure. This gives Equation 2.21 [69].

$$\boldsymbol{F} = \frac{\partial \tilde{\Pi}_{int}}{\partial \boldsymbol{\Gamma}} = \boldsymbol{C}_F \cdot \boldsymbol{\Gamma} \tag{2.21a}$$

$$\boldsymbol{M} = \frac{\partial \tilde{\Pi}_{int}}{\partial \boldsymbol{\Omega}} = \boldsymbol{C}_M \cdot \boldsymbol{\Omega} \tag{2.21b}$$

Above, $\boldsymbol{C}_F$ and $\boldsymbol{C}_M$ are constitutive tensors whose definition is given by Equation 2.22 [69].

$$\boldsymbol{C}_F = \begin{bmatrix} EA & 0 & 0 \\ 0 & G\bar{A}_2 & 0 \\ 0 & 0 & G\bar{A}_3 \end{bmatrix}_{\{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3\}} \tag{2.22a}$$

$$\boldsymbol{C}_M = \begin{bmatrix} GI_T & 0 & 0 \\ 0 & EI_2 & 0 \\ 0 & 0 & EI_3 \end{bmatrix}_{\{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3\}} \tag{2.22b}$$

In Equation 2.22, E is Young's modulus and G is the shear modulus. A is the cross-sectional area, and $\bar{A}_2$ and $\bar{A}_3$ are the two reduced cross-sectional areas. I_2 and I_3 are the principal moments of inertia, and I_T is the torsional moment of inertia.

The length-specific kinetic energy $\tilde{\Pi}_{kin}$ of the beam is given by Equation 2.23 [69].

$$\tilde{\Pi}_{kin} = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{c}_\rho \boldsymbol{w} + \frac{1}{2}\rho A \dot{\boldsymbol{r}}^T \dot{\boldsymbol{r}} \tag{2.23}$$

Above, ρ is the mass density, $\boldsymbol{c}_\rho$ is the spatial inertia tensor, and $\boldsymbol{w}$ is the spatial angular velocity. $\boldsymbol{c}_\rho$ is given by Equation 2.24.

$$\boldsymbol{c}_\rho = \frac{1}{2}\boldsymbol{\Lambda} \boldsymbol{C}_\rho \boldsymbol{\Lambda}^T \tag{2.24}$$

Here, $\boldsymbol{C}_\rho$ is the material inertia tensor. Its components are given by Equation 2.25.

$$\boldsymbol{C}_\rho = \begin{bmatrix} I_P & 0 & 0 \\ 0 & \rho I_2 & 0 \\ 0 & 0 & \rho I_3 \end{bmatrix}_{\{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3\}} \tag{2.25}$$

In Equation 2.25, I_P is the polar moment of inertia of the cross-section. To derive the length-specific linear momentum $\tilde{\boldsymbol{I}}$ and angular momentum $\tilde{\boldsymbol{h}}$, Equation 2.23 must be differentiated with respect to $\dot{\boldsymbol{r}}$ and $\boldsymbol{w}$, respectively. This results in Equation 2.26 [69].

$$\tilde{\boldsymbol{I}} := \frac{\partial \tilde{\Pi}_{kin}}{\partial \dot{\boldsymbol{r}}} = \rho A \dot{\boldsymbol{r}} \tag{2.26a}$$

$$\tilde{\boldsymbol{h}} := \frac{\partial \tilde{\Pi}_{kin}}{\partial \boldsymbol{w}} = \boldsymbol{\Lambda} \boldsymbol{C}_\rho \boldsymbol{W} \tag{2.26b}$$

The total quantities for the external forces, external moments, energies and momenta are found by integrating the distributed quantities along the length of the beam, resulting in Equation 2.27 [69].

$$\boldsymbol{f}_{ext} := \int_0^l \tilde{\boldsymbol{f}}\,\mathrm{d}s + \left[\bar{\boldsymbol{f}}\right]_{\Gamma_\sigma} \tag{2.27a}$$

$$\boldsymbol{m}_{ext} := \int_0^l \left(\boldsymbol{r}\times\tilde{\boldsymbol{f}} + \tilde{\boldsymbol{m}}\right)\,\mathrm{d}s + \left[\boldsymbol{r}\times\bar{\boldsymbol{f}} + \bar{\boldsymbol{m}}\right]_{\Gamma_\sigma} \tag{2.27b}$$

$$\Pi_{int} := \int_0^l \tilde{\Pi}_{int}\,\mathrm{d}s \tag{2.27c}$$

$$\Pi_{kin} := \int_0^l \tilde{\Pi}_{kin}\,\mathrm{d}s \tag{2.27d}$$

$$\boldsymbol{I} := \int_0^l \tilde{\boldsymbol{I}}\mathrm{d}s \tag{2.27e}$$

$$\boldsymbol{h} := \int_0^l \left(\tilde{\boldsymbol{h}} + \boldsymbol{r}\times\tilde{\boldsymbol{I}}\right)\mathrm{d}s \tag{2.27f}$$

Using Equation 2.26, the inertia force $\boldsymbol{f}_\rho$ and inertia moment $\boldsymbol{m}_\rho$ cab be derived as Equation 2.28 [69].

$$-\boldsymbol{f}_\rho := \dot{\tilde{\boldsymbol{I}}} = \rho A\ddot{\boldsymbol{r}} \tag{2.28a}$$

$$-\boldsymbol{m}_\rho := \dot{\tilde{\boldsymbol{h}}} = \boldsymbol{\Lambda}\left[\boldsymbol{S}(\boldsymbol{W})\boldsymbol{C}_\rho\boldsymbol{W} + \boldsymbol{C}_\rho\boldsymbol{A}\right] = \boldsymbol{S}(\boldsymbol{w})\boldsymbol{c}_\rho\boldsymbol{w} + \boldsymbol{c}_\rho\boldsymbol{a} \tag{2.28b}$$

Here, $\boldsymbol{a} := \dot{\boldsymbol{w}}$ and $\boldsymbol{A} = \dot{\boldsymbol{W}}$. It is possible to relate $\boldsymbol{w}$ and $\boldsymbol{a}$ to $\boldsymbol{\Psi}$ using Equation 2.29 [69].

$$\boldsymbol{w} = \boldsymbol{T}^{-1}\dot{\boldsymbol{\Psi}} \tag{2.29a}$$

$$\boldsymbol{a} = \dot{\boldsymbol{T}}^{-1}\dot{\boldsymbol{\Psi}} + \boldsymbol{T}^{-1}\ddot{\boldsymbol{\Psi}} \tag{2.29b}$$

Above, $\boldsymbol{T}$ is the tangent operator.

The last step is posing the boundary conditions and initial conditions for the problem so that one ends up with a well-defined initial boundary value problem. The boundary and initial conditions are given in Equation 2.30 [69].

$$\boldsymbol{r} = \boldsymbol{r}_u \quad \text{on } \Gamma_u \tag{2.30a}$$

$$\boldsymbol{\Psi} = \boldsymbol{\Psi}_u \quad \text{on } \Gamma_u \tag{2.30b}$$

$$\boldsymbol{f} = \bar{\boldsymbol{f}} \quad \text{on } \Gamma_\sigma \tag{2.30c}$$

$$\boldsymbol{m} = \bar{\boldsymbol{m}} \quad \text{on } \Gamma_\sigma \tag{2.30d}$$

$$\Gamma_u \cap \Gamma_\sigma = \emptyset \tag{2.30e}$$

$$\Gamma_u \cup \Gamma_\sigma = \{0, l\} \tag{2.30f}$$

$$\boldsymbol{r} = \boldsymbol{r}_0 \quad \text{at } t = 0 \tag{2.30g}$$

$$\dot{\boldsymbol{r}} = \boldsymbol{v}_0 \quad \text{at } t = 0 \tag{2.30h}$$

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_0 \quad \text{at } t = 0 \tag{2.30i}$$

$$\boldsymbol{w} = \boldsymbol{w}_0 \quad \text{at } t = 0 \tag{2.30j}$$

When these boundary and initial conditions are provided with the strong form Equation 2.14, then it is equivalent to the weak form Equation 2.15, provided that $\boldsymbol{r}(s,t)$ and $\boldsymbol{\Lambda}(\boldsymbol{\Psi}(s,t))$ have square-integrable first derivatives and satisfy Equation 2.30, and their variations are admissible and have squari-integrable first derivatives [69].

This concludes the literature review on Simo-Reissner beams. Compared to the Timoshenko and Euler-Bernoulli beams, the Simo-Reissner beam boasts many improvements such as the ability to handle large

deformations and arbitrary initial curvatures. Just like the Timoshenko beam, it is also able to capture shear deformation effects. In all, Simo-Reissner beams are extremely powerful elements with a large range of validity, but their one weakness is their introduction of extra unknowns. One must solve for 6 unknowns: $\boldsymbol{r}$ and $\boldsymbol{\Lambda}$ or $\boldsymbol{r}$ and $\boldsymbol{\Psi}$.

2.5.4. Geometrically nonlinear Kirchhoff rod theory

The overview of the geometrically nonlinear Kirchhoff beam and the formulae covered here were presented in a paper by Meier et al. (2015) on a locking free finite element formulation for geometrically exact Kirchhoff rods [80].

Although the theoretical basis of the Kirchhoff beam model was developed before the Simo-Reissner beam, Meier et al. (2015) noted that the state-of-the-art shear-free beam formulations based on the Kirchhoff theory had not yet attained the desirable properties of the geometrically exact Simo-Reissner formulation, such as objectivity. This formed the motivation for them to develop a geometrically exact, objective, large-deformation formulation for Kirchhoff beams [80].

As with the Simo-Reissner beam, first the kinematics of the geometrically nonlinear Kirchhoff beam are treated following the formulation presented by Meier et al. (2015) [80].

Kinematics

The starting point of the geometrically nonlinear Kirchhoff formulation is the same as the Simo-Reissner formulation. The geometrically nonlinear Kirchhoff beam uses the same arc-length parametrisation $s \mapsto \boldsymbol{r}(s) \in \mathbb{R}^3$ that was used by the Simo-Reissner beam in Subsection 2.5.3. As one can see in Figure 2.32, the undeformed configuration of the beam is defined by the space curve $s \mapsto \boldsymbol{r}_0(s) \in \mathbb{R}^3$. $s \in [0, L] \subset \mathbb{R}$ is the arc-length parameter of the curve, and $L \in \mathbb{R}$ is the undeformed length of the beam.

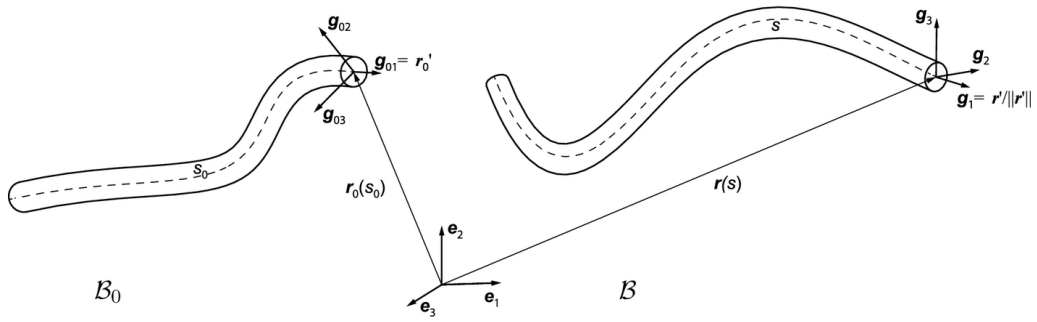


Figure 2.32: The kinematic quantities for the description a geometrically nonlinear Kirchhoff beam in its initial configuration, B_0, and current configuration, B, from [81].

A field of right-handed orthonormal base vectors $s \mapsto \boldsymbol{g}_{01}(s), \boldsymbol{g}_{02}(s), \boldsymbol{g}_{03}(s) \in \mathbb{R}^3$ completes the description of the initial configuration. These are fixed to the beam cross-sections, just as with Simo-Reissner beams. Similarly, the first base vector $\boldsymbol{g}_{01}(s)$ is the same as the unit tangent vector in the undeformed configuration, see Equation 2.31.

$$\boldsymbol{g}_{01}(s) \equiv \boldsymbol{r}_0'(s) \quad \text{with } ||\boldsymbol{r}_0'(s)|| = 1 \tag{2.31}$$

The base vectors $\boldsymbol{g}_{02}(s) and \boldsymbol{g}_{03}(s)$ point in the directions of the principal axes of inertia of the cross-section. Then, the orthogonal transformation $s \mapsto \boldsymbol{\Lambda}_0(s) \in SO(3)$ rotates the global frame $\boldsymbol{E}_i$ to the initial local frame $\boldsymbol{g}_{0i}(s)$ with Equation 2.32 [80].

$$\boldsymbol{g}_{0i}(s) = \boldsymbol{\Lambda}_0(s)\boldsymbol{E}_i(s) \quad \text{for } i = 1, 2, 3 \quad \text{with } \boldsymbol{\Lambda}_0(s) = \boldsymbol{g}_{0j}(s) \otimes \boldsymbol{E}_j \tag{2.32}$$

Assuming vanishing shear strains, which is a reasonable assumption for highly slender beams, one can introduce a constraint to the beam centreline that maintains each beam cross-section's orthogonality with the unit tangent $\boldsymbol{t}(s)/||\boldsymbol{t}(s)||$, with $\boldsymbol{t}(s) := \boldsymbol{r}'(s)$. This results in Equation 2.33 [80].

$$\boldsymbol{g}_2(s) \cdot \frac{\boldsymbol{t}(s)}{||\boldsymbol{t}(s)||} \equiv 0 \quad \text{and} \quad \boldsymbol{g}_3(s) \cdot \frac{\boldsymbol{t}(s)}{||\boldsymbol{t}(s)||} \equiv 0 \tag{2.33a}$$

$$\text{or} \quad \boldsymbol{g}_1(s) \equiv \frac{\boldsymbol{t}(s)}{||\boldsymbol{t}(s)||} \tag{2.33b}$$

Equation 2.33 is referred to as the Kirchhoff constraint. The next step is finding a parametrisation involving 4 degrees of freedom, $\big(\boldsymbol{r}(s), \phi(s)\big)$ with $s \mapsto \phi(s) \in \mathbb{R}$, that fulfils Equation 2.33 strongly. To this end, Meier et al. (2015) introduce a field of local right-handed orthonormal traids $s \mapsto \boldsymbol{g}_{ref,1}(s), \boldsymbol{g}_{ref,2}(s), \boldsymbol{g}_{ref,3}(s)$, which they refer to as reference triads. The reference triads are defined by the following conditions [80].

$$\boldsymbol{g}_{ref,1}(s) = \frac{\boldsymbol{r}'(s)}{||\boldsymbol{r}'(s)||} \tag{2.34a}$$

$$\boldsymbol{g}_{ref,2}(s) = \boldsymbol{f}_2(\boldsymbol{r}(s), \boldsymbol{r}'(s), \boldsymbol{r}''(s)) \tag{2.34b}$$

$$\boldsymbol{g}_{ref,3}(s) = \boldsymbol{f}_3(\boldsymbol{r}(s), \boldsymbol{r}'(s), \boldsymbol{r}''(s)) \tag{2.34c}$$

Here, the first base vector $\boldsymbol{g}_{ref,1}(s)$ is equal to the unit tangent vector to the centreline. The other reference triads can be arbitrary functions of primary variable $\boldsymbol{r}(s)$ and its derivatives. When the reference triads are rotated with respect to the tangent, one obtains the material triad [80], given by Equation 2.35.

$$\boldsymbol{g}_1(s) = \boldsymbol{g}_{ref,1} \tag{2.35a}$$

$$\boldsymbol{g}_2(s) = \boldsymbol{g}_{ref,2}(s) \cos\phi(s) + \boldsymbol{g}_{ref,3}(s) \sin\phi(s) \tag{2.35b}$$

$$\boldsymbol{g}_3(s) = \boldsymbol{g}_{ref,3}(s) \cos\phi(s) - \boldsymbol{g}_{ref,2}(s) \sin\phi(s) \tag{2.35c}$$

The fourth degree of freedom $\phi(s)$ introduced in Equation 2.35 is essentially an angle that represents the relative rotation between the reference and the material triad at a given s. Meier et al. (2015) define $\boldsymbol{g}_{ref,2}(s)$ and $\boldsymbol{g}_{ref,3}(s)$ more concretely later in their work, but for formulating the deformation measures, constitutive relations and the weak form, they state that the abstract expressions for the reference triad are enough [80].

The next step is to define the varation of the constrained rotation vector $\delta\boldsymbol{\theta}(s)$. This quantity is implicitly defined by Equation 2.36 [80].

$$\delta\boldsymbol{g}_i(s) = \delta\boldsymbol{\theta}(s) \times \boldsymbol{g}_i(s) \tag{2.36}$$

When the Kirchhoff constraint Equation 2.33 is used, one can obtain an expression for $\delta\boldsymbol{\theta}(s)$ which obeys the Kirchhoff constraint [80], see Equation 2.37.

$$\delta\boldsymbol{\theta}(s) = \delta\alpha(s)\frac{\boldsymbol{t}(s)}{||\boldsymbol{t}(s)||} + \frac{\boldsymbol{r}'(s) \times \delta\boldsymbol{r}'(s)}{\|\boldsymbol{r}'(s)\|^2} \tag{2.37}$$

Above, $s \mapsto \delta\boldsymbol{r}(s) \in \mathbb{R}^3$ describes the kinematically admissible variations of $\boldsymbol{r}(s)$. $\delta\alpha(s)$ is treated as an independent variational degree of freedom [80]. This term plays a key role later in Subsection 2.5.5.

Deformation measures and weak form

With the kinematics done, the deformation measures and the weak form can be described. From this point on, Meier et al. (2015) leave out the explicit dependence on s to make notation more concise. Once again, the first step towards posing the weak form is the strong form, given by Equation 2.38 [80].

$$\boldsymbol{f}' + \tilde{\boldsymbol{f}} = \boldsymbol{0} \tag{2.38a}$$

$$\boldsymbol{m}' + \boldsymbol{r}' \times \boldsymbol{f} + \tilde{\boldsymbol{m}} = \boldsymbol{0} \tag{2.38b}$$

In the equations above, $\tilde{\boldsymbol{f}}$ are the distributed external forces per unit length and $\tilde{\boldsymbol{m}}$ are the distributed external moments per unit length acting on the beam. $\boldsymbol{f}$ are the spatial force stress resulants and $\boldsymbol{m}$ are the spatial moment stress resultants.

Because shear strains are neglected, the shear forces will provide no work contribution. Therefore, to eliminate the shear forces from Equation 2.38, the force vector $\boldsymbol{f}$ is decomposed into a component parallel $\boldsymbol{f}_{\parallel}$ and a component perpendicular $\boldsymbol{f}_{\perp}$ to the tangent [80], see Equation 2.39.

$$\boldsymbol{f} = \boldsymbol{f}_{\parallel} + \boldsymbol{f}_{\perp} = f_{\parallel} \cdot \frac{\boldsymbol{t}}{||\boldsymbol{t}||} + \boldsymbol{f}_{\perp} \tag{2.39}$$

Here, $\boldsymbol{f}_{\parallel}$ is the axial force and $\boldsymbol{f}_{\perp}$ is the shear force. Using Equation 2.39, $\boldsymbol{f}_{\perp}$ can be eliminated from Equation 2.38, which results in a new system of equations in the unknowns $\boldsymbol{r}$ and ϕ given by Equation 2.40 [80].

$$\left(\frac{\boldsymbol{t}}{||\boldsymbol{t}||}\right)^T (\boldsymbol{m}' + \tilde{\boldsymbol{m}}) = 0 \tag{2.40a}$$

$$\boldsymbol{f}_{\parallel}' + \left[\frac{\boldsymbol{r}'}{||\boldsymbol{r}'||^2} \times (\boldsymbol{m}' + \tilde{\boldsymbol{m}})\right]' + \tilde{\boldsymbol{f}} = 0 \tag{2.40b}$$

The weak form can now be obtained by multiplying Equation 2.40a by the virtual twist rotation $\delta\alpha$ and Equation 2.40b by the virtual displacement $\delta\boldsymbol{r}$, and then integrating the result by parts twice along the beam. The resulting weak form can be seen in Equation 2.41a [80].

$$\begin{aligned} R(\boldsymbol{r}, \phi, \delta\alpha, \delta\boldsymbol{r}) = &\int_0^l \left[\delta\boldsymbol{\theta}'^T \boldsymbol{m} + \delta\boldsymbol{r}'^T \left(\frac{\boldsymbol{t}}{||\boldsymbol{t}||}\right) f_{\parallel} - \delta\boldsymbol{r}^T \tilde{\boldsymbol{f}} - \delta\boldsymbol{\theta}^T \tilde{\boldsymbol{m}}\right] \mathrm{d}s \\ &- \left[\delta\boldsymbol{r}^T \bar{\boldsymbol{f}} + \delta\boldsymbol{\theta}^T \bar{\boldsymbol{m}}\right]_{\Gamma_\sigma} = 0 \end{aligned} \tag{2.41a}$$

$$\delta\boldsymbol{\theta} = \delta\alpha \frac{\boldsymbol{t}}{||\boldsymbol{t}||} + \frac{\boldsymbol{r}' \times \delta\boldsymbol{r}'}{||\boldsymbol{r}'||^2} \tag{2.41b}$$

Equation 2.41a has been contracted so that all terms multiplying moments were grouped. The terms that form work conjugates with the moments, given by Equation 2.41b, turn out to be the variation of the constrained rotation vector from Equation 2.37. Γ_σ denotes the Neumann boundary, and the quantities $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{m}}$ are the external forces and moments on the Neumann boundary.

To obtain the deformation measures, one must invoke the principle of virtual work, which when applied to Equation 2.41a states that the terms multiplied by $\boldsymbol{m}$ and $f_{\parallel}$ must be the objective variation of the associated deformation measures, defined as $\boldsymbol{\omega}$ and ϵ. In equation terms: $\delta\boldsymbol{\theta}'^T \boldsymbol{m} = \delta_{rel}\boldsymbol{\omega}^T \boldsymbol{m}$ and $\delta\boldsymbol{r}'^T \left(\frac{\boldsymbol{t}}{||\boldsymbol{t}||}\right) f_{\parallel} = \delta\epsilon f_{\parallel}$, where the relative variation δ_{rel} of an arbitrary vector $\boldsymbol{a} \in \mathbb{R}^3$ means taking the objective part of the variation: $\delta_{rel}\boldsymbol{a} := \delta\boldsymbol{a} - \delta\boldsymbol{\theta} \times \boldsymbol{a}$.

Meier et al. (2015) then provide the deformation measures as Equation 2.42 [80].

$$\boldsymbol{\omega} = \begin{bmatrix} \tau + \phi' - \tau_0 - \phi_0' \\ (\boldsymbol{g}_2 \cdot \boldsymbol{\kappa}) - (\boldsymbol{g}_{02} \cdot \boldsymbol{\kappa}_0) \\ (\boldsymbol{g}_3 \cdot \boldsymbol{\kappa}) - (\boldsymbol{g}_{03} \cdot \boldsymbol{\kappa}_0) \end{bmatrix}_{\boldsymbol{g}_i} \tag{2.42a}$$

$$\epsilon = ||\boldsymbol{r}'|| - 1 \tag{2.42b}$$

Here, $\tau := \boldsymbol{g}'_{ref,2} \cdot \boldsymbol{g}_{ref,3} = -\boldsymbol{g}'_{ref,3} \cdot \boldsymbol{g}_{ref,2}$ describes the torsion of the reference triad. $\boldsymbol{\kappa}$ is related to the Frenet-Serret curvature, and it is defined by Equation 2.43 [80].

$$\boldsymbol{\kappa} := \frac{\boldsymbol{r}' \times \boldsymbol{r}''}{||\boldsymbol{r}'||^2} \tag{2.43}$$

With the deformation measures and weak form out of the way, this leaves the constitutive relations.

Constitutive relations
Since the Kirchhoff model assumes negligible shear strains, the constitutive relations shown in Subsection 2.5.3 can be simplified, particularly when it comes to the force stress resultants since the lack of shear strains reduces Equation 2.21a to a scalar equation. The constitutive relations, assuming hyperelastic material behaviour just as in Subsection 2.5.3, are given by Equation 2.44 [80].

$$f_{\parallel} = EA\epsilon \tag{2.44a}$$
$$\boldsymbol{m} = \boldsymbol{c}_m \boldsymbol{\omega} \tag{2.44b}$$

Here, E is Young's modulus and A is the cross-sectional area of the beam. The constitutive tensor $\boldsymbol{c}_m$ is defined as Equation 2.45 [80].

$$\boldsymbol{c}_m = \begin{bmatrix} GI_T & 0 & 0 \\ 0 & EI_2 & 0 \\ 0 & 0 & EI_3 \end{bmatrix}_{g_i} \tag{2.45}$$

In the equation above, G is the shear modulus, I_2 and I_3 are the principal moments of inertia, and I_T is the torional moment of inertia.

With this, the coverage of the geometrically nonlinear Kirchhoff rod formulation has come to a close. Overall, the geometrically nonlinear Kirchhoff beam is just as powerful a model as the Simo-Reissner beam. When it comes to utility, it is second only to the Simo-Reissner beam since the geometrcially nonlinear Kirchhoff formulation assumes negligible shear strains. However, this beam theory is equally capable of handling large deformations and arbitrary initial curvatures. One advantage it has over the Simo-Reissner formulation is the fact that one must only solve for 4 unknowns rather than 6. As such, when applied to slender beam problems, the geometrically nonlinear Kirchhoff formulation as presented by Meier et al. (2015) can be more computationally efficient than the Simo-Reissner formulation.

2.5.5. Torsion-free geometrically nonlinear Kirchhoff rod theory

The formulation treated in Subsection 2.5.4 is the most general formulation for geometrically nonlinear Kirchhoff beams capable of handling non-isotropic bending and arbitrary initial curvatures. In the following, a reduced formulation is treated, lowering both the complexity of the formulation and the numerical effort. One such reduced formulation is the torsion-free formulation [80].

Isotropic bending case
The torsion-free formulation of the geometrically nonlinear Kirchhoff rod as posed by Meier et al. (2015) is a subcase of the isotropic bending formulation. Because the torsion-free case is built on the isotropic bending case, the latter formulation is discussed quickly before moving on to the torsion-free formulation [80].

The starting point of the isotropic bending formulation is requiring that the problem is symmetric. The symmetry conditions are given by Equation 2.46 [80].

$$I_2 = I_3 =: I \tag{2.46a}$$
$$\boldsymbol{\kappa}_0 \equiv \boldsymbol{0} \tag{2.46b}$$

Here, $I_1 = I_2 = I$ implies the cross-section is quasi-circular, and $\boldsymbol{\kappa}_0 \equiv \boldsymbol{0}$ means vanishing initial curvature. This case is also known as the isotropic bending case. In this case, the undeformed state of the beam is rotationally symmetric around the centreline. The initial triad $\boldsymbol{g}_{0i}(s)$ can be chosen arbitrarily since any axis lying in the cross-sectional plane is a principal axis of inertia. For simplicity, Meier et al. (2015) chose a constant initial triad: $\boldsymbol{g}_{0i}(s) \equiv \boldsymbol{g}_{0i} = const.$, and as a result, the initial torsion goes to zero: $\tau_0 + \phi_0' \equiv 0$. Now, when the initial reference triad is the same as the triad $\boldsymbol{g}_{0i}$, then $\tau_0 \equiv 0$ and $\phi_0 \equiv 0$ [80].

When Equation 2.46, $\tau_0 \equiv 0$ and $\phi_0 \equiv 0$ are substituted into Equation 2.42a and Equation 2.44b, then the moment stress resultant for the isotropic beam formulation is given by Equation 2.47 [80].

$$\boldsymbol{m} = GI_T(\tau + \phi')\frac{\boldsymbol{t}}{||\boldsymbol{t}||} + EI\boldsymbol{\kappa} \tag{2.47}$$

Above, G is the shear modulus, I_T is the torsional moment of inertia, E is Young's modulus, and I is the principal moment of inertia.

As a result of the simplifications from the isotropic bending formulation, the bending moment $\boldsymbol{m}$ is no longer an explicit function of ϕ as seen in Equation 2.47. Now, Meier et al. (2015) propose the substitution $\alpha' := \tau + \phi'$, where α' signifies the total torsion. This way, they replace the primary variable ϕ with a different degree of freedom, α. Consequently, τ need not be calculated, and the weak form of this reduced formulation no longer depends on any reference triad. The simplifications result in the following weak form [80].

$$\begin{aligned} &R(\boldsymbol{r}, \alpha, \delta\boldsymbol{r}, \delta\alpha) = \\ &\int_0^l \left[\delta\boldsymbol{r}^T\tilde{\boldsymbol{f}} - \delta\boldsymbol{r}'^T\left(EA\boldsymbol{h_1} + EI\boldsymbol{h_2} + GI_T\boldsymbol{h_3} + \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||^2}\times\tilde{\boldsymbol{m}}\right) - \delta\boldsymbol{r}''^T EI\boldsymbol{h_4} + \delta\alpha\frac{\boldsymbol{r}'^T\tilde{\boldsymbol{m}}}{||\boldsymbol{r}'||} - \delta\alpha' GI_T\alpha'\right]\mathrm{d}s \\ &+ \left[\delta\boldsymbol{r}^T\bar{\boldsymbol{f}} - \delta\boldsymbol{r}'^T\left(\frac{\boldsymbol{r}'}{||\boldsymbol{r}'||^2}\times\bar{\boldsymbol{m}}\right) + \delta\alpha\frac{\boldsymbol{r}'^T\bar{\boldsymbol{m}}}{||\boldsymbol{r}'||}\right]_{\Gamma_\sigma} = 0 \end{aligned} \tag{2.48}$$

In Equation 2.48, $\boldsymbol{h_1}$, $\boldsymbol{h_2}$, $\boldsymbol{h_3}$ and $\boldsymbol{h_4}$ are shorthands defined by Equation 2.49.

$$\boldsymbol{h_1} := \boldsymbol{r}'\left(1 - \frac{1}{||\boldsymbol{r}'||}\right) \tag{2.49a}$$

$$\boldsymbol{h_2} := \frac{2\boldsymbol{r}'\left(\boldsymbol{r}'^T\boldsymbol{r}''\right)^2}{||\boldsymbol{r}'||^6} - \frac{\boldsymbol{r}'\left(\boldsymbol{r}''^T\boldsymbol{r}''\right) + \boldsymbol{r}''\left(\boldsymbol{r}'^T\boldsymbol{r}''\right)}{||\boldsymbol{r}'||^4} \tag{2.49b}$$

$$\boldsymbol{h_3} := \alpha'\frac{\boldsymbol{r}'\times\boldsymbol{r}''}{||\boldsymbol{r}'||^3} \tag{2.49c}$$

$$\boldsymbol{h_4} := \frac{\boldsymbol{r}''}{||\boldsymbol{r}'||^2} - \frac{\boldsymbol{r}'\left(\boldsymbol{r}'^T\boldsymbol{r}''\right)}{||\boldsymbol{r}'||^4} \tag{2.49d}$$

With this, the isotropic bending formulation has been covered. Compared to the general geometrically nonlinear Kirchhoff formulation the implementation complexity of this reduced formulation is lower owing to the fact that the weak form no longer depends on any reference triad. The weak form still has four unknowns, however, so the numerical cost will not be much lower than the general formulation. Furthermore, the isotropic bending case is only valid if the cross-section is circular and if there is no initial curvature. If one desires a way of making the geometrically nonlinear Kirchhoff rod formulation even more numerically efficient, then that is where the torsion-free formulation comes in. As such, the torsion-free case will next be treated in earnest.

Torsion-free case

To arrive at the torsion-free case, one more assumption is needed: the distributed and discrete external moments have no torsional components that act on the beam. This torsion-free constraint is given by Equation 2.50 [80].

$$\left(\frac{\boldsymbol{t}}{||\boldsymbol{t}||}\right)\cdot\tilde{m} = \tilde{m}_{||} \equiv 0 \tag{2.50a}$$

$$\left[\left(\frac{\boldsymbol{t}}{||\boldsymbol{t}||}\right)\cdot\bar{\boldsymbol{m}}\right]_{\Gamma_\sigma} = \left[\bar{m}_{||}\right]_{\Gamma_\sigma} = 0 \tag{2.50b}$$

Using the fact that the problem is symmetric, i.e., $I_1 = I_2 = I$ and $\boldsymbol{\kappa}_0 \equiv \boldsymbol{0}$, Meier et al. (2015) then use Equation 2.50a to show that the torsional component $m_{||}$ of the moment stress resultant $\boldsymbol{m}$ is constant in the torsion-free case [80]. This is expressed by Equation 2.51.

$$m'_{\parallel} = 0 \tag{2.51}$$

Combining Equation 2.51 with Equation 2.50b, one finds that $m_{\parallel}$ must be zero along the length of the beam, which is why this reduced formulation is known as torsion-free. Now, the degrees of freedom corresponding to the torsional deformation modes can be fully resigned. For the weak form, this means eliminating the torsional contributions, resulting in Equation 2.52 [80].

$$\begin{aligned} R(\boldsymbol{r}, \delta\boldsymbol{r}) = &\int_0^l \left[\delta\boldsymbol{r}^T \tilde{\boldsymbol{f}} - \delta\boldsymbol{r}'^T \left(EA\boldsymbol{h_1} + EI\boldsymbol{h_2} + \frac{\boldsymbol{r}'}{\|\boldsymbol{r}'\|^2} \times \tilde{\boldsymbol{m}} \right) - \delta\boldsymbol{r}''^T EI\boldsymbol{h_3} \right] \mathrm{d}s \\ &+ \left[\delta\boldsymbol{r}^T \bar{\boldsymbol{f}} - \delta\boldsymbol{r}'^T \left(\frac{\boldsymbol{r}'}{\|\boldsymbol{r}'\|^2} \times \bar{\boldsymbol{m}} \right) \right]_{\Gamma_\sigma} = 0 \end{aligned} \tag{2.52}$$

Here, E is Young's modulus and I is the moment of inertia. $\boldsymbol{h_1}$, $\boldsymbol{h_2}$, and $\boldsymbol{h_3}$ are shorthands for the following equations.

$$\boldsymbol{h_1} := \boldsymbol{r}' \left(1 - \frac{1}{||\boldsymbol{r}'||} \right) \tag{2.53a}$$

$$\boldsymbol{h_2} := \frac{2\boldsymbol{r}' \left(\boldsymbol{r}'^T \boldsymbol{r}'' \right)^2}{||\boldsymbol{r}'||^6} - \frac{\boldsymbol{r}' \left(\boldsymbol{r}''^T \boldsymbol{r}'' \right) + \boldsymbol{r}'' \left(\boldsymbol{r}'^T \boldsymbol{r}'' \right)}{||\boldsymbol{r}'||^4} \tag{2.53b}$$

$$\boldsymbol{h_3} := \frac{\boldsymbol{r}''}{||\boldsymbol{r}'||^2} - \frac{\boldsymbol{r}' \left(\boldsymbol{r}'^T \boldsymbol{r}'' \right)}{||\boldsymbol{r}'||^4} \tag{2.53c}$$

These expressions are essentially the same as Equation 2.49, except that there is no term dependent on α'. One will notice from Equation 2.52 that the weak form now only depends on $\boldsymbol{r}$ and its variations, meaning that only $\boldsymbol{r}$ needs to be discretised in numerical simulations.

In the torsion-free case, the constitutive law described in Subsection 2.5.4 (Equation 2.44) can be further simplified, now that torsional deformation has been resigned. Assuming the hyperelastic material behaviour, the constitutive relations now read [80]:

$$f_{\parallel} = EA\epsilon \tag{2.54a}$$

$$\boldsymbol{m}_{\perp} = EI\boldsymbol{\kappa} \tag{2.54b}$$

The only change is in the moment relation since there are now only bending moments $\boldsymbol{m}_{\perp}$. The constitutive tensor $\boldsymbol{c}_m$ is now a scalar, EI, because this value fully describes the cross-sectional bending stiffness of a cylindrical beam [80].

One final loose end is the variation of the constrained rotation vector, first posed as Equation 2.37. Meier et al. (2015) explain that in the isotropic bending case, the moment stress resultant no longer is a function of the reference triad or ϕ as mentioned above (Equation 2.47). With the torsion-free assumption, the variational degree of freedom $\delta\alpha$ is resigned, which means that $\delta\boldsymbol{\theta}$ can be simplified to Equation 2.55 [80] [16].

$$\delta\boldsymbol{\theta}(s) = \delta\boldsymbol{\theta}_{\perp}(s) := \frac{\boldsymbol{r}'(s) \times \delta\boldsymbol{r}'(s)}{\|\boldsymbol{r}'(s)\|^2} \tag{2.55}$$

Equation 2.55 essentially means that $\delta\boldsymbol{\theta}$ are the work conjugates of the bending moments $\boldsymbol{m}_{\perp}$ only.

This marks the end of the treatment of the torsion-free geometrically nonlinear Kirchhoff rod formulation. Compared to the general geometrically nonlinear Kirchhoff rod model, the torsion-free formulation has fewer unknowns since it only depends on the primary variables $\boldsymbol{r}$. This makes the reduced formulation more numerically efficient. On top of this, because the torsion-free formulation requires neither rotational degrees of freedom nor other triad parametrisations, it simplifies numerical procedures such as spatial discretisation, linearisation and time marching. In all, torsion-free geometrically nonlinear

Kirchhoff beams are even more computationally efficient than the general geometrically nonlinear Kirchhoff beam, and are extremely powerful tools if applied to a problem where the conditions outlined in Equation 2.50 are satisfied. The concessions, aside from the obvious resignation of the torsional modes, are that the initial curvature must be nill, and the cross-section must be circular.

2.5.6. Summary of the beam theories

This section has covered the formulation and characteristics of some of the more well-known beam theories from the literature. The first beam theory introduced was the Euler-Bernoulli beam, known for being very simple and efficient for small deflections and slender structures. As an extension of the Euler-Bernoulli theory, the Timoshenko formulation was covered next, which manages to capture the effects of shear deformations. After these simple beam formulations with limited validity, the Simo-Reissner formulation was treated, which has a level of utility nearly unmatched by other beam formulations. The general formulation can handle large deformations, arbitrary initial curvatures and shear deformation modes. However, it is also the most complex beam formulation as it requires solving for the full six unknowns. As a simplification of the Simo-Reissner formulation, the geometrically nonlinear Kirchhoff rod formulation foregoes shear deformations for computational efficiency. Finally, the torsion-free geometrically nonlinear Kirchhoff rod formulation, which assumes isotropic bending and no torsional deformation, can attain even higher computational efficiency at the cost of reducing the range of validity to symmetric problems without shear or torsional deformation modes.

2.6. Contact mechanics

The second component of the blueprint mentioned in Subsection 2.4.4 is modelling contact between struts using a beam-to-beam model. As with any contact model, one requires a contact formulation and a constraint enforcement technique [82]. However, before jumping ahead to the formulation of the beam-to-beam contact problem, it is worthwhile to consider the general contact problem, and the most common constraint enforcement approaches. After covering that, this section will pivot to the beam-to-beam contact problem.

2.6.1. The general contact problem

Formulating the general contact problem first is convenient because it makes explaining the contact enforcement methods easier when they are treated later. The theory and equations in the following were presented by Wriggers (2006) in his textbook on computational contact mechanics [82]. Figure 2.33 depicts a contact case between a solid and a rigid surface. The contact problem is identified as unilateral because the rigid surface constrains the motion of the body from one side.

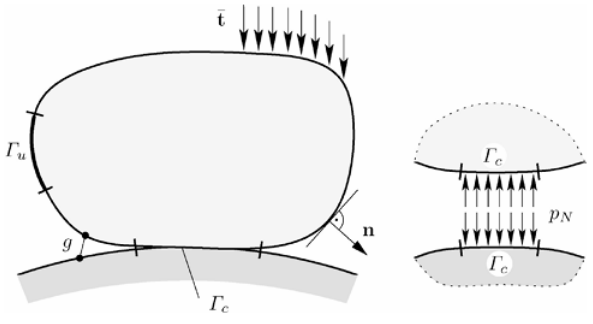


Figure 2.33: Unilateral contact of an elastic solid, from [82].

In Figure 2.33, $\bar{\boldsymbol{t}}$ is an applied traction that acts on the Neumann boundary Γ_σ (not shown in the figure) of the body $\mathcal{B}$, $\boldsymbol{n}$ is the outward normal vector of the surface of the body, Γ_u is the Dirichlet boundary, g is the gap function, Γ_c is the contact boundary, and p_N is the contact pressure [82].

Assuming linear elasticity, the equilibrium equation is given by Equation 2.56 [82].

$$-\text{div}\boldsymbol{\sigma} = \bar{\boldsymbol{f}} \quad \text{in } \mathcal{B} \tag{2.56}$$

Here, $\boldsymbol{\sigma}$ is the stress tensor at a point $\boldsymbol{X}$ inside of the body, and $\bar{\boldsymbol{f}}$ is the body force density. For the case of linear elasticity, the linear strain field $\boldsymbol{\epsilon}$ can be used to calculate the stress. The expression for $\boldsymbol{\epsilon}$ is given by Equation 2.57 [82].

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right) \tag{2.57}$$

One can see that $\boldsymbol{\epsilon}$ is a function of the displacement field $\boldsymbol{u}$. The stress is computed using Hooke's law for linear elasticity given by Equation 2.58.

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \mathbb{C}\boldsymbol{\epsilon}(\boldsymbol{u}) \tag{2.58}$$

Here, $\mathbb{C}$ is the elasticity tensor. Now, Wriggers poses the following boundary conditions [82].

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_u \tag{2.59a}$$
$$\boldsymbol{\sigma}\boldsymbol{n} = \bar{\boldsymbol{t}} \quad \text{on } \Gamma_\sigma \tag{2.59b}$$
$$u_N - g \leq 0 \quad \text{on } \Gamma_c \tag{2.59c}$$
$$p_N \leq 0 \quad \text{on } \Gamma_c \tag{2.59d}$$
$$(u_N - g)p_N = 0 \quad \text{on } \Gamma_c \tag{2.59e}$$

Above, Equation 2.59a are the Dirichlet conditions, Equation 2.59b are the Neumann conditions, and Equation 2.59c to Equation 2.59e are the contact conditions. One may recognise that Equation 2.59c is the non-penetration condition. The normal component of the displacement field is defined as $u_N = \boldsymbol{u}\cdot\boldsymbol{n}$, and the contact pressure is given by $p_N = \boldsymbol{t}\cdot\boldsymbol{n}$. This boundary value problem with inequalities associated with frictionless contact is called Signorini's problem [83].

The finite element solution of the boundary value problem requires the variational formulation of Signorini's problem. To this end, introduce a space $\mathcal{V}$ of vector-valued, real function defined on $\mathcal{B}$. Now, assume that the virtual displacements $\boldsymbol{v}$ obey the Dirichlet condition $\boldsymbol{v} = \boldsymbol{0}$ on Γ_u, and the contact condition $v_N - g \leq 0\ \forall\ \boldsymbol{v} \in \mathcal{V}$, where $v_N = \boldsymbol{v}\cdot\boldsymbol{n}$. With this, let $\boldsymbol{u}$ be the solution to Signorini's problem, and let the weak form of Equation 2.56 be given by Equation 2.60 [82].

$$\int_V \boldsymbol{\sigma}\cdot\boldsymbol{\epsilon}(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}V = \int_V \bar{\boldsymbol{f}}\cdot(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}V + \int_{\Gamma_\sigma} \bar{\boldsymbol{t}}\cdot(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}\Gamma + \int_{\Gamma_c} p_N(\boldsymbol{u})(u_N - v_N)\mathrm{d}\Gamma \tag{2.60}$$

According to Wriggers (2006), Equation 2.60 is equivalent to the virtual work done by the stresses $\boldsymbol{\sigma}$ and the virtual strains $\boldsymbol{\epsilon}(\boldsymbol{u}-\boldsymbol{v})$ as a result of the virtual displacements $\boldsymbol{u}-\boldsymbol{v}$ on one side of the equality sign, and the virtual work done by the external loads and body forces, on the other side of the equality sign [82].

The term in Equation 2.60 involving the contact pressure p_N can be reexpressed with Equation 2.59e to give Equation 2.61 [82].

$$p_N(u_N - v_N) = p_N(v_N - g) \geq 0 \tag{2.61}$$

Because of Equation 2.61, the solution of the Signorini problem must obey Equation 2.62.

$$\int_V \boldsymbol{\sigma}\cdot\boldsymbol{\epsilon}(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}V \geq \int_V \bar{\boldsymbol{f}}\cdot(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}V + \int_{\Gamma_\sigma} \bar{\boldsymbol{t}}\cdot(\boldsymbol{u}-\boldsymbol{v})\mathrm{d}\Gamma \tag{2.62}$$

One can see that the solution of Signorini's problem is described by a variational inequality as a result from the contact constraint. This is unlike normal problems in solid mechanics wherein solutions must satisfy variational equations. The presence of the inequality constraint on the deformation field turns the contact problem nonlinear, in spite of only considering a linear elastic problem.

Wriggers (2006) then extends the general contact formulation to two deformable bodies, but the structure is the same as for the single deformable body, with the weak form being an inequality. The main difference is that the contact domain contains both bodies now [82].

As a final note for the general contact problem, Figure 2.33 only considers frictionless contact. Adding friction into the equation means additional complexities since one must also model a unique constitutive response tangent to the contact interface. Frictional contact is characterised by transitions in contact behaviour from sticking, where tangential contact stresses are reaction stresses, to sliding, where the constitutive model is required to compute the tangential contact stresses. It is for this reason that frictional contact may even cause more complexities when it comes to the existence and uniqueness of frictional contact problems [82].

Examining the general contact case has already revealed that algorithms specially designed for contact problems are necessary to deal with the variational inequality. Subsection 2.6.3 will continue with that point, but before that, the general contact problem will be specified for hyperelastic materials.

2.6.2. Contact in hyperelastic materials

The purpose of the contact model is to capture beam-to-beam contact of lattice struts for impact problems. During impact, the struts can be expected to deform a lot, so the contact formulation must be compatible with finite deformations. As a preliminary to formulating the beam-to-beam contact problem, one can consider the case of two general hyperelastic bodies entering a state of contact. The equilibrium equations are given by Equation 2.63 [82].

$$\mathrm{div}\boldsymbol{P}^{\gamma} + \bar{\boldsymbol{f}}^{\gamma} = \boldsymbol{0} \tag{2.63}$$

Here, $\gamma = (1,2)$ for each body $\mathcal{B}^{\gamma}$. $\boldsymbol{P}^{\gamma}$ is the first Piola-Kirchhoff stress tensor acting in body γ, and $\bar{\boldsymbol{f}}^{\gamma} = \rho_0^{\gamma}\bar{\boldsymbol{b}}^{\gamma}$ are the body forces. Now, the boundary conditions for the deformation and stress fields are defined by Equation 2.64.

$$\phi^{\gamma} = \bar{\phi}^{\gamma} \quad \text{on } \Gamma_{\phi}^{\gamma} \tag{2.64a}$$

$$\boldsymbol{t}^{\gamma} = \bar{t}^{\gamma} \quad \text{on } \Gamma_{\sigma}^{\gamma} \tag{2.64b}$$

If contact is limited to frictionless contact, then the contact conditions on Γ_c are described by Equation 2.65 [82].

$$\begin{aligned} g_N &\geq 0 \\ p_N &\leq 0 \\ g_N p_N &= 0 \end{aligned} \tag{2.65}$$

The general expression for a hyperelastic constitutive law is given by Equation 2.66 [82].

$$\boldsymbol{P}^{\gamma} = \hat{\boldsymbol{P}}^{\gamma}\left(\boldsymbol{X}^{\gamma}, \boldsymbol{F}^{\gamma}, t\right) \tag{2.66}$$

Here, $\boldsymbol{X}$ is the position of a point in body $\mathcal{B}^{\gamma}$ in the reference configuration and $\boldsymbol{F}$ is the deformation gradient.

The weak form of the general two-body contact problem is obtained in a similar way to Equation 2.62. By using the boundary conditions prescribed in Equation 2.64 and Equation 2.65, the variational inequality can be found as Equation 2.67 [82].

$$\sum_{\gamma=1}^{2}\int_{V^{\gamma}} \boldsymbol{\tau}^{\gamma}\cdot\nabla\left(\boldsymbol{\eta}^{\gamma}-\boldsymbol{\phi}^{\gamma}\right)\mathrm{d}V \geq \sum_{\gamma=1}^{2}\int_{V^{\gamma}} \bar{\boldsymbol{f}}^{\gamma}\cdot\left(\boldsymbol{\eta}^{\gamma}-\boldsymbol{\phi}^{\gamma}\right)\mathrm{d}V - \int_{\Gamma_{\sigma}^{\gamma}} \bar{\boldsymbol{t}}^{\gamma}\cdot\left(\boldsymbol{\eta}^{\gamma}-\boldsymbol{\phi}^{\gamma}\right)\mathrm{d}\Gamma \tag{2.67}$$

Here, $\tau = \boldsymbol{P}\boldsymbol{F}^{T}$ is the Kirchhoff stress tensor. The deformation fields $\boldsymbol{\phi}^{\gamma} \in \boldsymbol{K}$ must now be found such that Equation 2.67 holds $\forall\, \boldsymbol{\eta}^{\gamma} \in \boldsymbol{K}$, where $\boldsymbol{K}$ is described by Equation 2.68 [82].

$$\boldsymbol{K} = \left\{ \left(\boldsymbol{\eta}^1, \boldsymbol{\eta}^2 \right) \in \mathcal{V} \mid \left[\boldsymbol{\eta}^2 - \hat{\boldsymbol{\eta}}^1 \left(\bar{\xi}^1, \bar{\xi}^2 \right) \bar{\boldsymbol{n}}_1 \right] \geq 0 \right\} \tag{2.68}$$

Equation 2.68 is true for arbitrary constitutive laws.

If a hyperelastic material is used, then the contact problem can be expressed as an optimisation problem with inequality constraints, see Equation 2.69 [82].

$$\frac{\partial \Pi^\gamma}{\partial \phi^\gamma} = 0 \quad \text{subject to } g_N \geq 0 \quad \text{on } \Gamma_c \tag{2.69}$$

The objective function to be minimised is the total potential energy, which for one body is given by Equation 2.70 [82].

$$\Pi^\gamma = \int_{V^\gamma} W^\gamma(\boldsymbol{C}) \mathrm{d}V - \int_{V^\gamma} \bar{\boldsymbol{f}}^\gamma \cdot \phi^\gamma \mathrm{d}V - \int_{\Gamma_\sigma^\gamma} \bar{\boldsymbol{t}}^\gamma \cdot \phi^\gamma \mathrm{d}A \tag{2.70}$$

Here, $W^\gamma(\boldsymbol{C})$ is the strain energy function of body $\mathcal{B}^\gamma$ and $\boldsymbol{C}$ is the right Cauchy-Green strain tensor.

Using the fact that the material if hyperelastic, Equation 2.67 can be reformulated as a minimisation problem, and under the assumption that the contact boundaries are known, the weak form becomes an equality. The total potential energy to be minimised comprises Equation 2.70 for each body, and a contribution from the contact constraints, Π_c. This leads to the weak form for two hyperelastic bodies in contact in Equation 2.71 [82].

$$\sum_{\gamma=1}^{2} \left\{ \int_{V^\gamma} W^\gamma(\boldsymbol{C}) \mathrm{d}V - \int_{V^\gamma} \bar{\boldsymbol{f}}^\gamma \cdot \phi^\gamma \mathrm{d}V - \int_{\Gamma_\sigma^\gamma} \bar{\boldsymbol{t}}^\gamma \cdot \phi^\gamma \mathrm{d}A \right\} + \Pi_c = 0 \tag{2.71}$$

Equation 2.71 is the basis for modelling frictionless contact in a mechanical system consisting of hyperelastic bodies. Now that a general contact formulation which is more applicable to the case of lattice struts, but not yet explicitly limited to beams, has been posed, it is time to cover how to enforce contact.

2.6.3. Enforcing contact constraints

Wriggers (2006) discusses many different methods for enforcing the contact constraints. Some methods have their roots in optimisation theory, while others come from mathematical programming. Contact problems in finite element formulations are generally solved with the Lagrange multiplier method or the penalty method [82]. More solutions methods are treated by Wriggers (2006) such as the Nitsche method, but this thesis will focus solely on the Lagrange multiplier and the penalty method.

Now follows a quick overview of the two enforcement methods. The Lagrange multipliers introduced in the Langrage multiplier method are treated as additional unknowns. Usually, said Lagrange multipliers, $\boldsymbol{\lambda}$, are introduced in the equation of motion, which means that the solution routine consists of solving the modified equation of motion and the contact (non-penetration) condition simultaneously for $\boldsymbol{\lambda}$, which are the contact forces, and the primary variables [84]. As one can imagine, solving the system of equations simultaneouly implies the contact constraints are enforced exactly and no interpenetration can occur.

In the penalty method, a penalty potential is added to the potential energy of the system. This penalty term results in a contact force proportional to the penetration. Through this penalty force, the contacting bodies are forced apart, which means contact is enforced approximately. Due to the nature of the penalty method, there must be interpenetration for there to be a contact force, so the penalty method allows some penetration, which means the contact constraint will be violated [82] [84].

Comparing the two methods, the Lagrange multiplier method has an advantage in that it prevents interpenetration altogether. The penalty method recovers the same solution as the Lagrange multiplier method in the limit where the contact penalty goes to infinity, however, very high contact penalty values adversely affect the stability of the solution. That said, one very good reason to prefer the

penalty method is the fact that it introduces no extra unknowns, meaning it is more computationally efficient than the Lagrange multiplier methods in problems with relatively few degrees of freedom [82] [17].

The Lagrange multiplier method and the penalty method are covered in more concrete terms in Subsection 2.6.4 and Subsection 2.6.5, respectively.

2.6.4. Lagrange multiplier method

To illustrate the Lagrange multiplier method for enforcing contact constraints, a simple example is taken from Wriggers (2006) [82]. Depicted in Figure 2.34 is a mass-spring system where a point mass m, subjected to gravitational load, is supported by a spring with stiffness k. The vertical movement of the mass is constrained by a rigid plane, and the distance between the mass and the rigid plane is h.

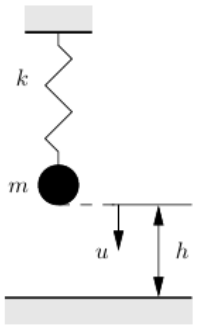


Figure 2.34: Point mass supported by a spring, from [82].

The energy of the mass-spring system is given by Equation 2.72 [82].

$$\Pi(u) = \frac{1}{2}ku^2 - mgu \tag{2.72}$$

The energy of the mass-spring system has been plotted in Figure 2.35. If u is not subjected to any constraints, then the optimum of Equation 2.72 can be calculated by variation, resulting in Equation 2.73 [82].

$$\delta\Pi(u) = ku\delta u - mg\delta u = 0 \tag{2.73}$$

The second variation of Π results in $\delta^2\Pi = k$, so the optimum of Equation 2.72 is a minimum at $u = \frac{mg}{k}$, which can be seen in Figure 2.35. The value of the energy at this minimum is denoted as Π_{min}.

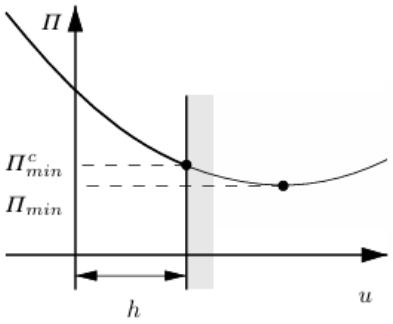


Figure 2.35: Energy of the mass spring system, from [82].

In actuality, the motion of the mass is constrained by the rigid surface below. This constraint on u is defined as Equation 2.74 [82].

$$c(u) = h - u \geq 0 \tag{2.74}$$

Equation 2.74 prohibits penetration through an inequality constraint. $c(u)$ represents the gap between the point mass and the rigid plane. For $c(u) = 0$, this gap is closed. Now, the variation δu at the contact surface is constrained. With Equation 2.74, this constraint is expressed as $\delta u \leq 0$. When this constrained variation is substituted in Equation 2.73, one obtains the variational inequality Equation 2.75 [82].

$$ku\delta u - mg\delta u \geq 0 \tag{2.75}$$

Because the constraint Equation 2.74 limits the permitted range of motion, the minimum of Equation 2.72 is not necessarily Π_{min} with $u = \frac{mg}{k}$. Instead, the constrained minimum of Equation 2.72, Π^c_{min}, is found at a different, admissible u. This constrained minimum can also be seen in Figure 2.35.

The solution of u is governed by the variational inequality Equation 2.75. It is not possible to solve the contact problem directly with the variational inequalities, instead a dedicated solution method is needed, such as the Lagrange multiplier method.

Assuming that there is contact, then the Lagrange multiplier method augments Equation 2.72 with a term that involves the constraint Equation 2.74. This gives Equation 2.76 [82].

$$\Pi(u, \lambda) = \frac{1}{2}ku^2 - mgu + \lambda c(u) \tag{2.76}$$

As can be seen in Figure 2.36, λ is equivalent to the contact force. Taking the first variation of Equation 2.76 results in two equations because λ is an independent unknown like u. The system of equations is given by Equation 2.77 [82].



Figure 2.36: The contact force represented by a Lagrange multiplier, from [82].

$$ku\delta u - mg\delta u - \lambda\delta u = 0 \tag{2.77a}$$
$$c(u)\delta\lambda = 0 \tag{2.77b}$$

Equation 2.77a describes the equilibrium of the point mass when it interacts with the rigid plane. Equation 2.77b indicates that the solution satisfies the kinematic constraint Equation 2.74. Now, the variation is not constrained anymore, meaning one can solve for λ, and find that it is equal to the reaction force at the contact surface [82].

Going beyond the simple problem presented by Wriggers (2006), to apply the Lagrange multiplier method to a finite element model, one must first consider the semi-discretised equation of motion of the

finite element system, given by Equation 2.78 [84].

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{R} \tag{2.78}$$

Above, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{F}$ is the internal force vector, $\boldsymbol{R}$ is the external force vector, and $\boldsymbol{x}$ is the degree of freedom vector.

Now, let there be a set of active contact constraints which prescribe the displacements to prevent interpenetration between contacting bodies. These constraints can be defined as Equation 2.79 [84].

$$\boldsymbol{G}\boldsymbol{x} = \boldsymbol{0} \tag{2.79}$$

Here, $\boldsymbol{G}$ is a surface contact displacement constraint matrix. It acts on $\boldsymbol{x}$ to ensure no penetration occurs when there is contact.

To solve this contact problem, the Lagrange multipliers are introduced in the equation of motion as the vector $\boldsymbol{\lambda}$, resulting in the constrained equation of motion Equation 2.80 [84].

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{R} \tag{2.80}$$

The components of $\boldsymbol{\lambda}$ are the surface contact forces. The solution to the contact problem is obtained by solving Equation 2.80 and Equation 2.79 simultaneously [84].

It should be said that the formulation provided by Carpenter et al. (1991) [84] implicitly assumes the contact constraint is linear, so that it can be expressed using the matrix $\boldsymbol{G}$. More generally, the contact constraint is represented by the vector-valued function given by Equation 2.81.

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0} \tag{2.81}$$

Here, $\boldsymbol{g}$ does the same thing as $\boldsymbol{G}$ from Equation 2.79, it operates on the degrees of freedom and ensures that there is no penetration. Since there is no more $\boldsymbol{G}$ matrix, the Lagrange multipliers need to be introduced slightly differently into the equation of motion, see Equation 2.82.

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{F}_{\text{contact}}(\boldsymbol{\lambda}) = \boldsymbol{R} \tag{2.82}$$

The contact force vector $\boldsymbol{F}_{\text{contact}}$ is a function of the Lagrange multiplier vector $\boldsymbol{\lambda}$. $\boldsymbol{\lambda}$ is not necessarily the same dimension as the degree of freedom vector $\boldsymbol{x}$, so its contribution to each degree of freedom is mapped by $\boldsymbol{F}_{\text{contact}}$. The solution to this contact problem is obtained in a similar manner as before, one must solve the system of two equations, Equation 2.82 and Equation 2.81, simultaneously.

From this point on, the solution becomes rather specific to the time marching scheme employed, so the coverage stops at this point, however, one can see that the Lagrange multiplier method extends easily to finite element problems.

2.6.5. Penalty method

To explain how the penalty method approaches contact problems, consider the same mass-spring system introduced in Subsection 2.6.4, shown in Figure 2.34. When the contact constraint Equation 2.74 is active, the penalty method modifies the total energy Equation 2.72 by adding a penalty term, see Equation 2.83 [82].

$$\Pi(u) = \frac{1}{2}ku^2 - mgu + \frac{1}{2}\epsilon\left[c(u)\right]^2, \quad \epsilon > 0 \tag{2.83}$$

Here, ϵ is the penalty parameter. It can be seen as the spring stiffness associated with the rigid support when it is in contact with the point mass [82]. This has been illustrated in Figure 2.37.

When contact is active, the variation of Equation 2.83 results in Equation 2.84 [82].

$$ku\delta u - mg\delta u - \epsilon c(u)\delta u = 0 \tag{2.84}$$

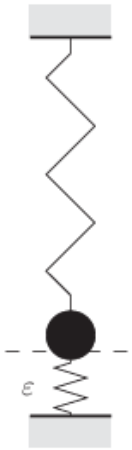


Figure 2.37: The point mass supported by a spring and a penalty spring due to the penalty term, from [82].

The solution of Equation 2.84 can be found to be Equation 2.85 [82].

$$u = \frac{mg + \epsilon h}{k + \epsilon} \tag{2.85}$$

When this solution is substituted into the constraint equation Equation 2.74, one obtains the expression below [82].

$$c(u) = h - u = \frac{kh - mg}{k + \epsilon} \tag{2.86}$$

Because $mg \geq kh$ when contact occurs, Equation 2.86 implies the point mass penetrates into the rigid support [82]. So just as explained in Subsection 2.6.3, the penalty method will allow some penetration. This penetration is a function of the penalty parameter ϵ, and the penetration $c(u)$ will only be 0 in the limit where ϵ goes to infinity. In other words, the exact solution, which is the solution for which the contact constraint Equation 2.74 is satisfied, is only obtained when the penalty stiffness is infinitely high [82].

The contact force is then defined as a function of the penalty parameter and the penetration distance, see Equation 2.87 [82].

$$f = \epsilon c(u) = \frac{\epsilon}{k + \epsilon}(kh - mg) \tag{2.87}$$

The contact force and the solution u obtained using the penalty method converge to the contact force and the displacement found using the Lagrange multiplier method in the limit $\epsilon \to \infty$ [82]. In this limit, the support becomes rigid and the contact problem becomes the same as Figure 2.36. The fact the solution with the penalty method converges to the solution with the Lagrange multiplier method means the penalty method enforces the contact constraint approximately.

When applying the penalty method to contact problems in finite element systems, one can take a similar approach as outlined in Subsection 2.6.4. The semi-discrete equation of motion Equation 2.78 and the contact constraints Equation 2.81 are the same. The penalty method then also introduces contact forces into the equation of motion, leading to Equation 2.88.

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{F}_{\text{contact}}(\epsilon, \boldsymbol{x}) = \boldsymbol{R} \tag{2.88}$$

The difference with Equation 2.82 is that the contact force $\boldsymbol{F}_{\text{contact}}$ is now a function of the penalty parameter ϵ and the degrees of freedom $\boldsymbol{x}$. The reason for the latter dependency is because the penalty contact force is proportional to the penetration $\boldsymbol{g}$, which itself is a function of $\boldsymbol{u}$. Because ϵ is chosen a-priori, the solution to the contact problem does not involve solving Equation 2.88 and Equation 2.81 simultaneously, but rather, only Equation 2.88 needs to be solved since there are no extra unknowns.

Now, it must be said that choosing a suitable penalty parameter for a finite element simulation is not trivial. Given the knowledge that the exact contact solution is recovered when $\epsilon \to \infty$, one may naively

choose a very high penalty parameter. In the case of the mass-spring example Figure 2.34, one can get away with simply choosing an extremely high ϵ since there is no time stepping or multiple degrees of freedom involved. But for finite element systems, large values for ϵ make the system stiffer, which adversely affects the numerical stability and forces the use of smaller time steps [84]. Since choosing too high of a penalty value causes numerical instability, and choosing too low of a penalty value risks excessive interpenetration or even failed contact detection, finding the optimal penalty value is usually a matter of tuning ϵ through trial and error. That said, understanding of the physics of the problem can aid in choosing an optimal penalty value.

2.6.6. Formulating the beam-to-beam contact problem

As promised, now that the general contact problem has been covered, and the two most common contact enforcement methods have been explained, the formulation for the beam-to-beam contact problem can be treated. Because beams are parametrised as 1D curves in space, the beam-contact problem is formulated differently from the general formulation treated in Subsection 2.6.1. Since beam contact is essentially a matter of finding contact between curves, the kinematic conditions for contact are formulated differently [82].

The beam-to-beam contact formulation covered in the following was treated in detail by Wriggers and Zavarise (1997), and by Meier et al. (2016) [85] [17]. An example of a beam-contact problem can be seen in Figure 2.38.

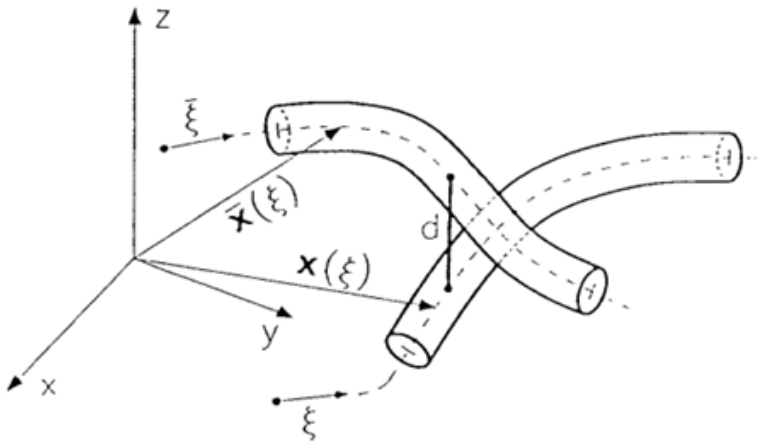


Figure 2.38: A general beam-contact problem, from [85].

Here, ξ and $\bar{\xi}$ are the curve parameters for each beam. d denotes the closest distance between the two beams, which is to say, the closest distance between the two curves. Wriggers (2006) states that as long as the beams are not parallel and have smooth, convex cross-sections, contact between the two curves results in one contact point locally [82]. As such, the contact formulation treated below is referred to by Meier et al. (2016) as a point-to-point contact formulation [17].

Point-to-point contact formulation

To illustrate the point-to-point contact formulation, let there be two beams of arbitrary configuration. Their cross-sectional radii are R_1 and R_2, respectively, and their centrelines are given by two parametric curves $\boldsymbol{r}_1(\xi)$ and $\boldsymbol{r}_2(\eta)$, with $\xi \in [-1, 1]$ and $\eta \in [-1, 1]$. Additionally, $\boldsymbol{r}_{1,\xi}(\xi)$ and $\boldsymbol{r}_{2,\eta}(\eta)$ are the tangents of said curves at ξ and η. The abovementioned kinematic quantities can also be seen in Figure 2.39 [17].

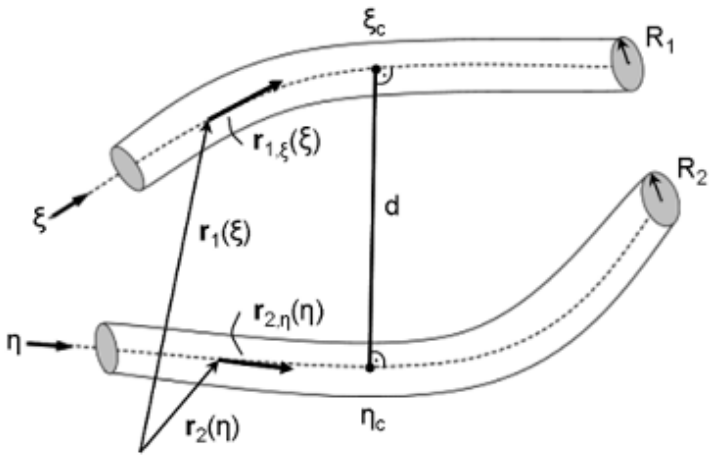


Figure 2.39: Kinematic quantities for the point-to-point contact problem, from [17].

In Figure 2.39, ξ_c and η_c denotes the closest points of the two beams. The subscript c means the quantity is evaluated at ξ_c or η_c. To find these closest point coordinates, one must solve the bilateral minimum distance problem, defined as Equation 2.89 [17].

$$d_{bl} := \min_{\xi,\eta} d(\xi,\eta) = d(\xi_c,\eta_c) \quad \text{with } d(\xi,\eta) = ||\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)|| \tag{2.89}$$

The above problem results in two orthogonality conditions, which are used to find the solution ξ_c and η_c [17].

$$\begin{aligned} p_1(\xi,\eta) &= \boldsymbol{r}_{1,\xi}^T(\xi)\left[\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)\right] \rightarrow p_1(\xi_c,\eta_c) \doteq 0 \\ p_2(\xi,\eta) &= \boldsymbol{r}_{2,\eta}^T(\eta)\left[\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)\right] \rightarrow p_2(\xi_c,\eta_c) \doteq 0 \end{aligned} \tag{2.90}$$

From here, the contact constraint preventing interpenetration at the closest points is defined by Equation 2.91.

$$g \geq 0 \quad \text{with } g := d_{bl} - (R_1 + R_2) \tag{2.91}$$

Above, g is the gap function.

Because the orthogonality conditions in Equation 2.90 are, in general, nonlinear in ξ and η, finding the solution to said conditions involves an iterative solver like a Newton-Raphson scheme. When the orthogonality conditions from Equation 2.90 are linearised with respect to ξ and η, one obtains the system of equations depicted by Equation 2.92 [85].

$$\begin{aligned} &\begin{bmatrix} -\boldsymbol{r}_{1,\xi}(\xi)\cdot\boldsymbol{r}_{1,\xi}(\xi) + \left[\boldsymbol{r}_2(\eta) - \boldsymbol{r}_1(\xi)\right]\cdot\boldsymbol{r}_{1,\xi\xi}(\xi) & \boldsymbol{r}_{1,\xi}(\xi)\cdot\boldsymbol{r}_{2,\eta}(\eta) \\ -\boldsymbol{r}_{1,\xi}(\xi)\cdot\boldsymbol{r}_{2,\eta}(\eta) & \boldsymbol{r}_{2,\eta}(\eta)\cdot\boldsymbol{r}_{2,\eta}(\eta) + \left[\boldsymbol{r}_2(\eta) - \boldsymbol{r}_1(\xi)\right]\cdot\boldsymbol{r}_{2,\eta\eta}(\eta) \end{bmatrix} \begin{bmatrix} \Delta\xi_c \\ \Delta\eta_c \end{bmatrix} \\ &= -\begin{bmatrix} \left[\boldsymbol{r}_2(\eta) - \boldsymbol{r}_1(\xi)\right]\cdot\boldsymbol{r}_{1,\xi}(\xi) \\ \left[\boldsymbol{r}_2(\eta) - \boldsymbol{r}_1(\xi)\right]\cdot\boldsymbol{r}_{2,\eta}(\eta) \end{bmatrix} \end{aligned} \tag{2.92}$$

When Equation 2.92 is solved, one obtains the solution increment for ξ_c and η_c for iteration $n+1$. The update equation for the solution is then $\xi_c^{n+1} = \xi_c^n + \Delta\xi_c^{n+1}$, and $\eta_c^{n+1} = \eta_c^n + \Delta\eta_c^{n+1}$ [85].

With the contact points known, the constraint can be incorporated into the problem by defining a penalty potential contribution to the variational problem, see Equation 2.93 [17].

$$\Pi_{c\epsilon} = \frac{1}{2}\epsilon\langle g\rangle^2 \quad \text{and} \quad \langle x\rangle = \begin{cases} x, & \text{if } x \leq 0 \\ 0, & \text{if } x > 0 \end{cases} \tag{2.93}$$

Here, ϵ is the penalty parameter.

When the Lagrange multiplier method is used instead of the penalty method for enforcing contact, then the contribution to the variational problem is described by Equation 2.94 [17].

$$\begin{aligned} \Pi_{c\lambda} &= \lambda g \\ \lambda &\geq 0 \\ g &\geq 0 \\ \lambda g &= 0 \end{aligned} \tag{2.94}$$

Above, λ is the Lagrange multiplier to be solved. However, Meier et al. (2016) primarily treat the penalty method in their work, so the following will also mainly focus on the penalty method for the beam-contact problem [17]. The reason why they prefer the penalty method will be explained later.

Taking the first variation of Equation 2.93 yields the contribution of one contact point to the weak form, Equation 2.95 [17].

$$\delta\Pi_{c\epsilon} = \epsilon\langle g\rangle(\delta \boldsymbol{r}_{1c} - \boldsymbol{r}_{2c})^T \boldsymbol{n} \tag{2.95}$$

From Equation 2.95, the contact force and the normal vector can be defined as Equation 2.96 and Equation 2.97 [17].

$$\boldsymbol{f}_{c\epsilon} = -\epsilon\langle g\rangle \boldsymbol{n} = f_{c\epsilon}\boldsymbol{n} \tag{2.96}$$

$$\boldsymbol{n} = \frac{\boldsymbol{r}_1(\xi_c) - \boldsymbol{r}_2(\eta_c)}{||\boldsymbol{r}_1(\xi_c) - \boldsymbol{r}_2(\eta_c)||} \tag{2.97}$$

Equation 2.96 states that in the point-to-point contact formulation, the contact force $\boldsymbol{f}_{c\epsilon}$ acting on two beams in contact are represented as discrete point forces applied to ξ_c and η_c.

With this, the required background for implementing a beam-to-beam contact model using the point-to-point contact formulation has been covered. However, one may have noticed the use of the phrase "as long as the beams are not parallel" earlier when the point-contact formulation was first introduced. Indeed, it turns out that the point-contact formulation has limits, which Meier et al. (2016) elaborate on in detail in the following [17].

Limitations of the point-to-point contact formulation
According to Meier et al. (2016) the point-to-point contact formulation is an elegant and efficient formulation if the contact angles are sufficiently large. It is this latter part which points to the limitation of the point-contact model, since a unique closest point solution to Equation 2.90 cannot be guaranteed for arbitrary configurations [17].

Meier et al. (2016) then present a procedure for determining the bounds on certain control quantities for which a unique closest point solution can be guaranteed. The control quantities in question are the contact angle α (Equation 2.98), the closest point distance d_{bl} and the geometric curvature $\bar{\kappa}$ of the beam's centreline (Equation 2.99) [17].

$$\alpha = \arccos(z) \quad \text{with } z = \frac{||\boldsymbol{r}_1'^T(\xi_c)\boldsymbol{r}_2'(\eta_c)||}{||\boldsymbol{r}_1'(\xi_c)||||\boldsymbol{r}_2'(\eta_c)||},\ \alpha \in [0; 90°] \tag{2.98}$$

α is the angle between the tangent vectors at the contact point.

$$\bar{\kappa} := \frac{\kappa}{\boldsymbol{r}'} = \frac{||\boldsymbol{r}' \times \boldsymbol{r}''||}{||\boldsymbol{r}'||^3} = ||\boldsymbol{r}_{,\tilde{s}\tilde{s}}|| \quad \text{since } ||\boldsymbol{r}_{,\tilde{s}}|| = 1 \tag{2.99}$$

Above, $\tilde{s} \in [0; \tilde{l}]$ is the arc length parameter in the current configuration, and $\tilde{l}$ is the deformed length. Now, let the maximum cross-sectional to curvature radius μ_{max} be defined as Equation 2.100 [17].

$$\mu_{max} = \frac{R}{min(\bar{r})} \ll 1 \quad \text{with } \bar{r} = \frac{1}{\bar{\kappa}} \tag{2.100}$$

Here, R is the cross-sectional radius, and $\bar{r}$ is the radius of curvature. According to Meier et al. (2016), $\mu_{max} \ll 1$ must hold for problems modelled by beams. The next step is defining the unilateral distance function field $d_{ul}(\tilde{s}_1)$. This field maps a given point $\tilde{s}_1$ on the slave beam to a closest partner point $\tilde{s}_{2c}$ of the master beam. This unilateral closest point mapping is defined by Equation 2.101 [17] (also illustrated in Figure 2.41).

$$d_{ul}(\tilde{s}_1) = \min_{\tilde{s}_2} d(\tilde{s}_1, \tilde{s}_2) = d(\tilde{s}_1, \tilde{s}_{2c}) \quad \text{with } d(\tilde{s}_1, \tilde{s}_2) = ||\boldsymbol{r}_1(\tilde{s}_1) - \boldsymbol{r}_2(\tilde{s}_2)|| \tag{2.101}$$

Now, it can be observed that bilateral closest point projection (Equation 2.89) is a special case of unilateral closest point projection (Equation 2.101). The closest point coordinates from Equation 2.89 are then found by minimising $d_{ul}(\tilde{s}_1)$ with respect to $\tilde{s}_1$, see Equation 2.102 [17].

$$d_{bl} = \min_{\tilde{s}_1} d_{ul}(\tilde{s}_1) = d_{ul}(\tilde{s}_{1c}) \tag{2.102}$$

From here, Meier et al. (2016) rigorously derive the requirements for the existence and uniqueness of a solution to the bilateral closest point problem. They first prove how to guarantee existence and uniqueness for the unilateral distance problem, after which they derive the condition for a unique bilateral closest point projection. For the mathematical details, one can consult the paper by Meier et al. (2016) [17]. Their results are also shown below.

Assuming the two beams are near enough so that the closest point $\tilde{s}_{2c}$ is relevant when it comes to active contact constraints ($g(\tilde{s}_1) = 0 \rightarrow d_{ul}(\tilde{s}_1) = 2R$), the requirement for a unique solution to the unilateral distance problem is given by Equation 2.103 [17].

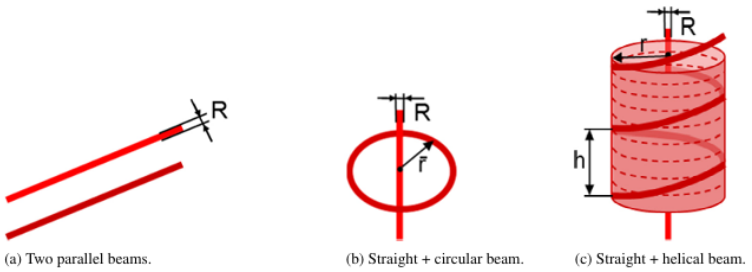
$$2\frac{R}{\bar{r}(\tilde{s}_{2c})} \leq 2\mu_{max} \dot{<} 1 \tag{2.103}$$

Because of Equation 2.100, uniqueness of $d_{ul}(\tilde{s}_1)$ can be guaranteed if the beams are close enough.

The requirement for a unique bilateral closest point projection is defined as a lower bound on the contact angle, see Equation 2.104 [17].

$$\alpha \dot{>} \alpha_{min} = \arccos(1 - 2\mu_{max}) \tag{2.104}$$

From this condition, one can see that providing an upper bound for μ_{max}, that is, ensuring existence of the unilateral distance solution, automatically results in a lower bound for the contact angles for which the bilateral distance problem has a unique solution [17]. Meier et al. (2016) provide three examples to put Equation 2.104 into a practical context, see Figure 2.40.



(a) Two parallel beams. (b) Straight + circular beam. (c) Straight + helical beam.

Figure 2.40: Some examples of contact problems with non-unique bilateral distance solutions, from [17].

Figure 2.40 (a) showcases two straight beams, meaning $\mu_{max} = 0$. In this case, the requirement becomes $\alpha > 0$, which is exactly what is expected to obtain a unique closest point solution for two parallel beams. Figure 2.40 (b) illustrates the case of a straight beam and a circular beam centred on the straight beam. Here, the gap $g(\tilde{s}_1)$ can be observed to be constant between the two beams, since all points on the circular beam are equidistant to one point on the straight beam (the point that coincides with the circular beam's centre). Therefore, there is no unique bilateral closest point solution, even if the contact

angle is far from 0° at $\alpha = 90°$. The third example, Figure 2.40 (c), depicts contact between a straight beam and a helical beam. In this case, the gap function $g(\tilde{s}_1)$ is also constant, which again leads to no unique bilateral closest point solution. In a sense, the helix is the middleground between the two previous examples. For $h \to \infty$, one ends up with two parallel straight lines, and for $h = 0$, one gets the case of a straight beam and a circular beam.

Line-to-line contact formulation

Depending on μ_{max}, the bilateral distance problem may not have a unique solution even for non-parallel beams when the contact angle is greater than 0°. It is for this reason that Meier et al. (2016) propose a line-to-line contact formulation. Instead of searching for one closest pair of points, this model looks for a closest point field $\eta_c(\xi)$ on the master beam as a function of the parametric coordinate field ξ on the slave beam. The kinematic quantities of this formulation are depicted in Figure 2.41 [17].

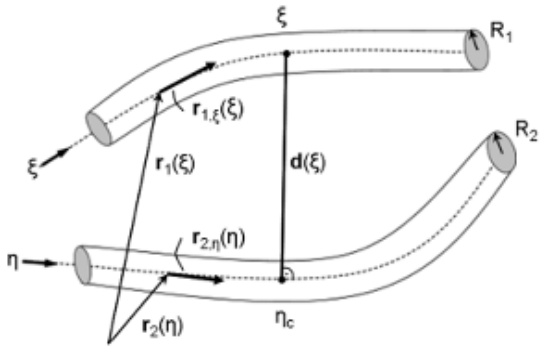


Figure 2.41: Kinematic quantities for the line-to-line contact problem, from [17].

The nearest master point η_c to a given slave point ξ is the solution of the unilateral minimum distance problem Equation 2.105 [17].

$$d_{ul}(\xi) := \min_{\eta} d(\xi, \eta) = d(\xi, \eta_c) \quad \text{with } d(\xi, \eta) = ||\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)|| \tag{2.105}$$

Equation 2.105 results in a single orthogonality condition which is to be solved for η_c [17].

$$p_2(\xi, \eta) = \boldsymbol{r}_{2,\eta}^T(\eta)(\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)) \to p_2(\xi, \eta_c) \doteq 0 \tag{2.106}$$

Unlike with point-contact, in the case of line-contact, the normal vector is only perpendicular to the master beam, but it is no longer necessarily perpendicular to the slave beam. Now, in the context of line-contact, the subscript c signifies that a quantity is evaluated at the closest master point η_c of a given slave point ξ. The contact constraint is then defined as Equation 2.107 [17].

$$g(\xi) \geq 0 \; \forall \; \xi \quad \text{with } g(\xi) := d_{ul} - R_1 - R_2 \tag{2.107}$$

The constraint above is formulated along the entire slave beam, hence the $\forall \; \xi$. This gap function only gives the exact value of the minimal beam line-to-line distance at a given coordinate ξ if the contact normal vector is orthogonal to the centrelines of both beams (Equation 2.108) [17].

$$\boldsymbol{r}_{1,\xi}^T(\xi)\boldsymbol{n}(\xi) = 0 \quad \text{and} \quad \boldsymbol{r}_{2,\eta}^T(\eta_c)\boldsymbol{n}(\xi) = 0 \tag{2.108}$$

For the point-to-point contact formulation, the above conditions are exactly satisfied at the closest point by definition. However, for the line-to-line formulation, only the second condition holds for any ξ. That said, for $\epsilon \to \infty$, the line-contact formulation actually converges to the point-contact formulation [17].

The choice of constraint enforcement by Meier et al. (2016) is based on the space-continuous penalty potential (Equation 2.109) [17].

$$\Pi_{c\epsilon} = \frac{1}{2}\epsilon \int_0^{l_1} \langle g(\xi) \rangle^2 ds_1 \tag{2.109}$$

When the first variation of Equation 2.109 to taken, one finds the contribution to the weak form as described in Equation 2.110 [17].

$$\delta\Pi_{c\epsilon} = \epsilon \int_0^{l_1} \langle g(\xi)\rangle \delta g(\xi) ds_1 \quad \text{and} \quad \delta g(\xi) = (\delta \boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\xi))^T \boldsymbol{n}(\xi) \tag{2.110}$$

The contact force $\boldsymbol{f}_{c\epsilon}$ and the normal vector $\boldsymbol{n}(\xi)$ can be recovered from the variation form Equation 2.110.

$$\boldsymbol{f}_{c\epsilon}(\xi) = -\epsilon\langle g(\xi)\rangle \boldsymbol{n}(\xi) = f_{c\epsilon}(\xi)\boldsymbol{n}(\xi) \tag{2.111}$$

$$\boldsymbol{n}(\xi) := \frac{\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta_c)}{||\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta_c)||} \tag{2.112}$$

Equation 2.111 states that the contact force $\boldsymbol{f}_{c\epsilon}$ is modelled as a distributed line force by the line-to-line contact model. From here on, Meier et al. (2016) cover the spatial discretisation of the line-to-line contact formulation and the subsequent solution procedure. One of the key features of line-contact is that contact must be discretised too, since one deals with a distributed penalty force. Compare this with the point-contact case, where no such discretisation is necessary. If the contact region consists of one master element and one slave element, then Meier et al. (2016) give their discretised contribution to the weak form as Equation 2.113 [17].

$$\delta \boldsymbol{d}_1^T \sum_{k=1}^{n_{GP}} w_k J(\xi_k)\epsilon\langle g(\xi_k)\rangle \boldsymbol{N}_1^T(\xi_k)\boldsymbol{n}(\xi_k) + \delta \boldsymbol{d}_2^T \sum_{k=1}^{n_{GP}} -w_k J(\xi_k)\epsilon\langle g(\xi_k)\rangle \boldsymbol{N}_2^T(\eta_c(\xi_k))\boldsymbol{n}(\xi_k) \tag{2.113}$$

In the above equation, n_{GP} is the number of quadrature points per slave element, w_k are the associated quadrature weights, ξ_k are the coordinates of the quadrature points in the parametric domain $\xi \in [-1;1]$, and lastly, $\eta_c(\xi_k)$ is the closest master point to the quadrature point ξ_k, see Figure 2.42. $J(\xi_k)$ is the Jacobian which describes the transformation from the current configuration to the parametric space. $\boldsymbol{N}$ represents the matrix of shape functions for the elements, and $\boldsymbol{d}$ are the degree of freedom vectors of each element. One can also identify the residual contributions of each element, $\boldsymbol{r}_{con,1}$ and $\boldsymbol{r}_{con,2}$, as the vector which multiplies the variation of the degree of freedoms $\delta\boldsymbol{d}$ in Equation 2.113 [17].

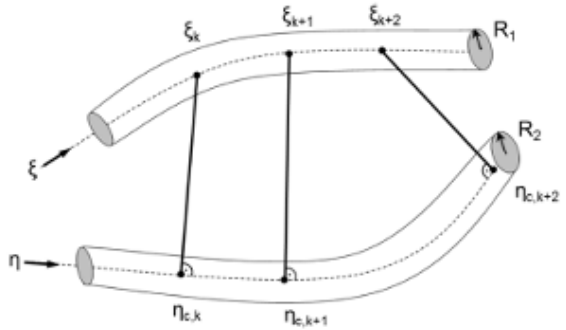


Figure 2.42: Kinematic quantities for the discretised line-to-line contact problem, from [17].

Meier et al. (2016) then point out that it is preferable to decouple the discretisation of the beam from the discretisation of contact. What they propose is to allow $n_{II} \geq 1$ contact integration subdomains per slave beam element with n_{GR} quadrature points on each of these subdomains, which results in $n_{GP} = n_{II} \cdot n_{GR}$ quadrature points per slave element. To achieve this, one must introduce n_{II} more parametric spaces $\bar{\xi}_i \in [-1;1]$ with $i = 1, ..., n_{II}$ to the slave element, see Equation 2.114 [17].

$$\xi(\bar{\xi}_i) = \frac{1.0 - \bar{\xi}_i}{2}\xi_{1,i} + \frac{1.0 + \bar{\xi}_i}{2}\xi_{2,i} \quad \text{with } i = 1, ..., n_{II} \tag{2.114}$$

The total Jacobian can be derived from Equation 2.114, resulting in Equation 2.115 [17].

$$J(\xi(\bar{\xi}_i)) = \frac{ds_1}{d\bar{\xi}_i} = \frac{\partial s_1}{\partial \xi}\frac{\partial \xi}{\partial \bar{\xi}_i} \quad \text{with } i = 1, ..., n_{II} \tag{2.115}$$

Now, the residual contributions of each element can be expressed as Equation 2.116 [17].

$$\boldsymbol{r}_{con,1} = \sum_{i=1}^{n_{II}} \sum_{j=1}^{n_{GR}} w_j J(\xi_{ij}, \xi_{1,i}, \xi_{2,i}) \epsilon \langle g(\xi_{ij}) \rangle \boldsymbol{N}_1^T(\xi_{ij}) \boldsymbol{n}(\xi_{ij}) = \boldsymbol{r}_{con,1}^{ij} \tag{2.116a}$$

$$\boldsymbol{r}_{con,2} = \sum_{i=1}^{n_{II}} \sum_{j=1}^{n_{GR}} -w_j J(\xi_{ij}, \xi_{1,i}, \xi_{2,i}) \epsilon \langle g(\xi_{ij}) \rangle \boldsymbol{N}_2^T(\eta_c(\xi_{ij})) \boldsymbol{n}(\xi_{ij}) = \boldsymbol{r}_{con,2}^{ij} \tag{2.116b}$$

In Equation 2.116, the residual contributions $\boldsymbol{r}_{con,1}^{ij}$ and $\boldsymbol{r}_{con,2}^{ij}$ are of one quadrature point j in the integration interval i. Also, the parametric coordinates ξ_{ij} are evaluated using Equation 2.114 at the quadrature point coordinates $\bar{\xi}_j$, see Equation 2.117 [17].

$$\xi_{ij} = \frac{1.0 - \bar{\xi}_j}{2} \xi_{1,i} + \frac{1.0 + \bar{\xi}_j}{2} \xi_{2,i} \quad \text{with } i = 1, ..., n_{II},\ j = 1, ..., n_{GR} \tag{2.117}$$

The solution scheme is more involved than for point-contact, so for the details, one can consult Appendix C of Meier et al. (2016) [17]. The final result is given below.

$$\boldsymbol{k}_{con,l}^{ij} = \frac{d\boldsymbol{r}_{con,l}^{ij}}{d\boldsymbol{d}_{12}} = \frac{\partial \boldsymbol{r}_{con,l}^{ij}}{\partial \boldsymbol{d}_{12}} + \frac{\partial \boldsymbol{r}_{con,l}^{ij}}{\partial \eta_c} \frac{\partial \eta_c}{\partial \boldsymbol{d}_{12}}, \quad l = 1, 2 \tag{2.118}$$

This is the linearised contribution of one quadrature point on element l to the residual. This means that unlike with point-contact, where the closest point is found by solving a system of equations separate from the equation of motion, line-contact couples the solution of the unilateral minimum distance problem to the discretised equation of motion [17].

In all, the line-to-line contact formulation provides more robustness than the point-to-point formulation since it can handle arbitrary contact angles owing to the fact that it need only solve a unilateral minimum distance problem. However, a numerical implementation of a line-contact model is noticeably more involved than a point-contact model.

Penalty or Lagrange multipliers

The choice of constraint enforcement technique has already been briefly touched upon in the previous, and one may have noticed a preference for the penalty method on the part of Meier et al. (2016) [17]. This is not without reason. In the following, the penalty method and the Lagrange multiplier method as applied to the beam-contact problem are compared to each other.

The Lagrange multiplier method introduces an additional primary variable field λ which is usually discretised similarly to the displacement field. These nodal variables of the discretised Lagrange multiplier field are extra unknowns, and solving the augmented system involves mortar techniques [17], which makes the implementation more involved compared to the penalty method. Meier et al. (2016) further explain that the Lagrange multiplier method is very commonly applied in computational contact mechanics for solids because of their higher accuracy of contact resolution, which is a big advantage of the Lagrange multiplier method. However, they remark that majority of publications on beam-contact problems employ the penalty method [17].

One of the reasons why penalty methods are preferred over the Lagrange multiplier method in the case of beam-contact problems has to do with with the ratio of Lagrange multiplier degrees of freedom to the displacement degrees of freedom of a finite element system. For beams, the ratio of surface degrees of freedom to all degrees of freedom is 1, which is much higher than for 3D continuum elements. As a result, the share of the Lagrange multiplier degrees of freedom of the total number of degrees of freedom is relatively large for beam-to-beam contact problems. This means the system to be solved at each time step is a lot larger because of the introduction of the Lagrange multipliers, making the solution more computationally expensive [17].

2.6.7. Summary of contact mechanics review

This section was dedicated to the second component of the unified framework, the contact model. The basis of the contact formulation was explained first in the most general setting, after which it was specified to hyperelastic materials to better reflect the eventual beam-contact scenario. Then, the two foremost constraint enforcement methods were introduced in general terms: the penalty and Lagrange multiplier method. They were illustrated using a simple example provided by Wriggers (2006) [82].

The final and most salient part of this section treated the beam-to-beam contact formulation by Wriggers and Zavarise (1997) [85], and Meier et al. (2016) [17]. There, the two approaches to formulating the beam contact problem were discussed: the point-to-point formulation and the line-to-line formulation. Both formulations have their strengths and weaknesses, which will be used in Subsection 3.1.5 to come to a proper choice of formulation. Also treated in the context of the beam-contact problem were the two contact enforcement techniques mentioned earlier. In this application, the penalty method is preferred over the Lagrange multiplier method because of the comparatively lower computational cost and the relative simplicity.

2.7. Fracture mechanics

The last component of the blueprint mentioned in Subsection 2.4.4 is modelling damage in the struts of a lattice metamaterial using a fracture model. Unlike the damage evolution models mentioned in Subsection 2.4.3, the sought-after fracture model should capture fracture without needing to resort to element deletion. Before the discussing the fracture model applied to the case of beams, it pays to discuss the general field of computational fracture mechanics. After that, one of the most useful fracture models is discussed in a more general setting, before it is applied to the beam setting.

2.7.1. Computational fracture mechanics

Computational fracture mechanics can be applied to a broad spectrum of computational methods to model fracture problems, but the foremost application case is in the finite element method. The very first attempt to model fracture in FEM came in the form of elements based on linear elastic fracture mechanics (LEFM), followed by elastic-plastic fracture mechanics (EPFM). These methods at first only performed well when non-uniform mesh refinement was applied, but it did not take long before uniform meshes could also work fine. Rapid developments in computational fracture mechanics soon brought about more advanced techniques such as the extended finite element method (XFEM) [86].

2.7.2. Cohesive zone model

A well studied and often used fracture model is the cohesive zone model (CZM) [87]. It offers key advantages over linear elastic fracture mechanics in that it can handle non-negligible nonlinear zones ahead of the crack tip, and it needs no initial crack [88]. Cohesive zone models are generally characterised by a cohesive constitutive relationship between the displacement discontinuity and the cohesive traction. This constitutive relationship is often modelled using a traction-separation law. Some key features of this cohesive law are summarised below [87].

- The work required to initiate a crack surface is finite, its value equal to the fracture energy (the area under the traction-separation curve).
- There is a finite characteristic length scale, which means that prolonged loading eventually leads to decohesion, that is, fracture.
- The cohesive traction on the fracture surface generally decreases to 0 as the separation increases under softening.

Elices et al. (2002) remark that the cohesive model only contains information on the softening part of the load-displacement curve, where progressive damage in the form of cracks initiate and grow. The hardening part of the load-displacement curve, where fracture has not initiated yet, must come from the continuum outside of the cohesive zone. This, they refer to as the bulk [88]. This fact makes applying the cohesive zone model to FEM as interface elements between bulk elements quite intriguing since the two types of elements complement each other. Where the bulk elements model 'regular' (non-fracture) behaviour, the cohesive zone interface elements capture the effects related to fracture such as softening

and eventual failure.

The cohesive law, which Elices et al. (2002) also call the softening function, is considered a material property. The softening function is defined and characterised by a number of parameters, which include the cohesive strength, the fracture energy, the characteristic length and the critical crack length. The characteristic length is essentially an inverse measure of the brittleness of the material. The way these parameters affect softening behaviour can be illustrated in two cases. If a cracked structure is much larger than the cohesive zone, then the fracture energy fully determines the peak force. If the cracked material is not too large and if the structure is not cracked, then the cohesive strength and the critical crack length determine the structural response [88].

Since cohesive zones pair up quite well with the bulk material in a finite element context, they are a prime candidate for modelling fracture in beams. Beam elements will only share at most two interfaces with their neighbours, which are in the cross-section plane. Because beam elements are represented with 1D curves, this makes incorporating the cohesive zone model into a finite element model simpler and less computationally intensive since the fracture surface is not explicitly modelled, but rather it is the beam cross-section.

2.7.3. Modelling fracture in beams

The formulation treated here was presented by Kota et al. (2024) in their study on a discontinuous Galerkin/cohesive zone model formulation for fracture in geometrically exact slender beams [16]. Before discussing the application of the cohesive zone model to beams, one should first investigate the fracture modes of beams. These fracture modes have been depicted in Figure 2.43.

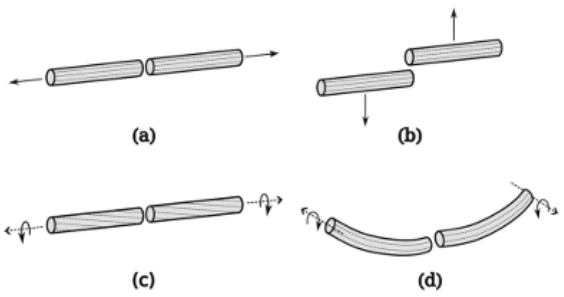


Figure 2.43: The fracture modes of beams: (a) tension, (b) shear, (c) torsion and (d) bending, from [16].

One can see that beams have four failure modes: under tension, due to shear, because of torsion and caused by bending. Kota et al. (2024) proceed to neglect the shear and torsional fracture modes, keeping only the tensile and bending fracture modes for slender beams [16].

The next step is formulating the cohesive law. As explained in Subsection 2.7.2, the cohesive law comes in the form of a traction-separation law. Since the cohesive zone model is applied to beams, Kota et al. (2024) derive the traction-separation law using stress resultants. Furthermore, it is assumed that cohesive forces and moments are only functions of the kinematic discontinuities near the crack tip [16].

Now, the interfaces of neighbouring beam elements, where there may be jumps in the kinematic fields, are given cohesive boundaries. Here, Kota et al. (2024) set the following definitions for the axial and bending kinematic discontinuities [16].

$$\begin{aligned} \Delta_{||} &= [\![\boldsymbol{r}]\!] \cdot \hat{\boldsymbol{n}}_{coh} \\ \boldsymbol{\Theta} &= [\![\boldsymbol{g}_1]\!] \end{aligned} \tag{2.119}$$

Here, $\boldsymbol{r}$ and $\boldsymbol{g}_1$ are as defined in Subsection 2.5.3 and Subsection 2.5.4. $\hat{\boldsymbol{n}}_{coh}$ is the unit normal vector to the cohesive interface in the deformed state,see Equation 2.120 [16].

$$\hat{\boldsymbol{n}}_{\boldsymbol{coh}} = \frac{\langle \boldsymbol{g_1} \rangle}{||\langle \boldsymbol{g_1} \rangle||} \tag{2.120}$$

In the above equations, and in the equations to follow later in this thesis, the following notations are used.

$$[\![\bullet]\!]\Big|_{s=a} := \lim_{s \to a^+} \bullet - \lim_{s \to a^-} \bullet \tag{2.121}$$

$$\langle \bullet \rangle\Big|_{s=a} := \frac{1}{2}\left(\lim_{s \to a^+} \bullet + \lim_{s \to a^-} \bullet \right) \tag{2.122}$$

Equation 2.121 signifies a jump in the field $\bullet$ and Equation 2.122 denotes the average of the field $\bullet$ at some point $s \in (0, L)$. Kota et al. (2024) point out that the component of $\boldsymbol{\Theta}$ in the direction of $\hat{\boldsymbol{n}}_{coh}$ is zero because $\boldsymbol{\Theta} \cdot \hat{\boldsymbol{n}}_{coh} = 0$ holds by construction [16].

The effect of tension and bending on fracture can be represented by a single quantity: the effective separation (Equation 2.123). This allows for capturing mixed-mode fracture due to tension and bending [16].

$$\Delta = \sqrt{\{\Delta_{||}\}^2 + \left(\alpha R ||\boldsymbol{\Theta}||\right)^2}, \tag{2.123}$$

Here, R is the radius of the beam and α is a mode-mixity parameter. Furthermore, $\{\bullet\} = \max(\bullet, 0)$ is the Macaulay operator [16].

The cohesive axial forces and cohesive bending moments associated with the aforementioned kinematic jumps are defined as Equation 2.124 [16].

$$\begin{aligned} \boldsymbol{f_{coh,||}} &= f_{coh}(\Delta, \boldsymbol{q}) \, \frac{\{\Delta_{||}\}}{\Delta} \, \hat{\boldsymbol{n}}_{\boldsymbol{coh}}, \\ \boldsymbol{m_{coh,\perp}} &= \alpha^2 \, f_{coh}(\Delta, \boldsymbol{q}) \, \frac{R^2}{\Delta} \, \boldsymbol{\Theta}, \end{aligned} \tag{2.124}$$

Above, f_{coh} is the effective cohesive force, which is a function of the effective separation Δ and some internal variables $\boldsymbol{q}$. Depending on what kind of cohesive behaviour is to be modelled, such as brittle or ductile, $f_{coh}(\Delta, \boldsymbol{q})$ can be specified accordingly. Kota et al. (2024) assume a linear relation between the cohesive axial forces and cohesive bending moments and Δ. With this cohesive model, increasing Δ causes a linearly proportional reduction in f_{coh}. To capture the fact that damage is permanent, they also introduce a history variable Δ_{max} as an internal variable. Δ_{max} is defined as the maximum effective separation encountered throughout the loading history [16]. This constitutive fracture behaviour is illustrated in Figure 2.44.

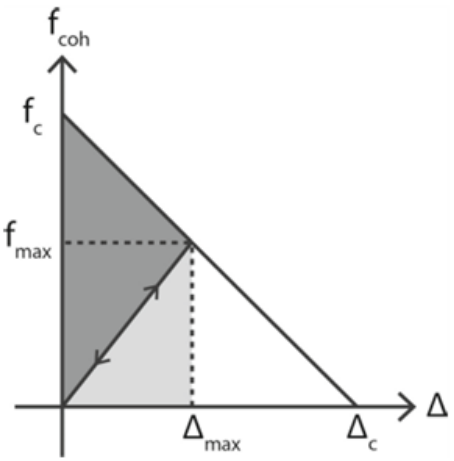


Figure 2.44: A linear traction-separation law for modelling beam fracture, from [16].

In Figure 2.44, one can observe two loading paths. During loading, f_{coh} follows a trajectory from f_c to 0 as Δ increases to Δ_c. During unloading, the effective cohesive force follows the path to the origin, and during reloading, f_{coh} follows that same path back to the loading state for Δ_{max}. Here, $f_c = \sigma_c A$ signifies the critical value for the effective cohesive force, with σ_c being the cohesive strength of the material and A the beam's cross-sectional area. $\Delta_c = \frac{2G_c}{\sigma_c}$ is the effective separation at which fracture occurs, with G_c as the effective fracture energy of the material. The cohesive law for loading is given by Equation 2.125 [16].

$$f_{coh}(\Delta, \Delta_{max}) = \left(1 - \frac{\Delta}{\Delta_c}\right) f_c \quad \text{for} \quad \Delta \geq \Delta_{max} \tag{2.125}$$

The unloading and reloading path is defined by Equation 2.126 [16].

$$f_{coh}(\Delta, \Delta_{max}) = \frac{\Delta}{\Delta_{max}} f_{max} \quad \text{for} \quad \Delta < \Delta_{max} \tag{2.126}$$

In the equation above, f_{max} is defined as f_{coh} at Δ_{max}. Both Equation 2.125 and Equation 2.126 are only active in an interface when the fracture initiation criterion below is met [16].

$$f_{eq}(\langle \boldsymbol{f} \rangle, \langle \boldsymbol{m}_\perp \rangle) \geq f_c, \tag{2.127}$$

In Equation 2.127, f_{eq} is an equivalent force defined in Equation 2.128 [16].

$$f_{eq}(\boldsymbol{f}, \boldsymbol{m}_\perp) = \sqrt{\{\boldsymbol{f} \cdot \hat{\boldsymbol{n}}_{coh}\}^2 + \left\| \frac{\boldsymbol{m}_\perp}{\alpha R} \right\|^2}. \tag{2.128}$$

Here, $\boldsymbol{f}$ is the internal force, and $\boldsymbol{m}_\perp$ is the internal bending moment.

2.7.4. Summary of fracture mechanics review

The main purpose of this section was introducing the fracture model which will be used in the unified framework for beam contact and fracture. Computational fracture mechanics has come a long way since the early days of applying linear elastic fracture mechanics to FEM, and one of the state-of-the-art fracture models is the cohesive zone model. The cohesive zone model uses a constitutive relation to govern the fracture process. One advantage offered by CZM over linear elastic fracture mechanics is the fact that no initial crack is required to analyse fracture.

When applied to the case of beam fracture, the model used by Kota et al. (2024) consists of a linear traction-separation law (Equation 2.125 and Equation 2.126) determined by material properties such as σ_c and G_c. For unloading and reloading, a history variable Δ_{max} is also used to represent permanent

damage. This cohesive law acts as the constitutive law of an interface element when the fracture initiation criterion (Equation 2.127) is met [16].

3

Methodology

One of the prerequisites of modelling a large-scale truss-metamaterial problem that captures both fracture and self-contact is finding a way to integrate beam contact behaviour with fracture, hence why this was also formulated as a subtask. On top of this, modelling impact in large scale truss-lattice structures also involves modelling decisions pertaining to joints, contact with the impactor and parallelisation, to name a few. All of this warrants in-depth exposition, which is why the current chapter covers the methodology followed.

The layout of the methodology is as follows. Section 3.1 is dedicated to synthesising a computational framework which unifies beam contact with fracture behaviour. This framework will fully describe the dynamics of the struts, but modelling lattice structures requires more than just describing the behaviour of the struts. To this end, Section 3.2 covers the steps involved in modelling large-scale lattice structures for impact problems. Then, Section 3.3 presents a way of quantifying energy absorption mechanisms by tracking energy components. Lastly, Section 3.4 gives an overview of the SUMMIT framework which provides the basis for the simulations and the computational framework developed in this study.

3.1. Synthesis of a unified contact and beam fracture framework

Subtask 1

Incorporate beam-contact behaviour with beam-fracture behaviour, and tie it with the strut dynamics.

The first topic to cover is developing a unified inter-element contact and beam fracture framework, where beam-fracture behaviour is integrated with self-contacting behaviour of beams. To this end, this section explains how the theoretical building blocks covered in the second half of Chapter 2 have been applied to realise such a framework. The key steps involved are summarised below.

Subsection 3.1.1 and Subsection 3.1.2 cover the choice of beam model and the formulation of the governing equation. Then, Subsection 3.1.3 presents the weak form and offers a way of incorporating fracture behaviour into the bulk beam behaviour. Afterwards, the discretisation of the problem is treated in Subsection 3.1.4. This is followed by the implementation details of the beam-contact formulation and its incorporation into the beam dynamics (Subsection 3.1.5, Subsection 3.1.6, Subsection 3.1.7 and Subsection 3.1.8). The section is closed out with the key assumptions that went into this framework.

3.1.1. Choice of beam model for truss architected materials

After reviewing the most relevant state-of-the-art beam models for slender structures in Section 2.5, one can now list and compare the main features of each beam theory. The criteria factoring into the comparison are whether the beam model (1) is nonlinear, (2) works with arbitrary initial curvatures ($\kappa_0 \neq 0$), (3) is formulated for anisotropic cross-sections ($I_2 \neq I_3 \neq I_1$), (4) has a low computational cost,

(5) is relatively simple to implement, and whether the formulation can capture the deformation modes of (6) shear and (7) torsion. A summary of the beam models and their fulfilment of each criterion is presented in Table 3.1 below.

Table 3.1: Overview of beam models (partially adapted from [69])

Formulations	Theory	$\kappa_0 \neq 0$	$I_2 \neq I_3 \neq I_1$	Computational cost	Simplicity	Shear	Torsion
Euler-Bernoulli	Linear	-	+	+	+	-	+
Timoshenko	Linear	-	+	+	+	+	+
Simo-Reissner	Nonlinear	+	+	-	-	+	+
Geometrically nonlinear Kirchhoff	Nonlinear	+	+	+/-	-	-	+
Symmetric geometrically nonlinear Kirchhoff	Nonlinear	-	-	+/-	+	+	+
Torsion-free geometrically nonlinear Kirchhoff	Nonlinear	-	-	+	+	-	-

Some notes about Table 3.1, a "+" signifies the formulation fulfils the criterion, a "−" means it does not fulfil the criterion and "+/−" indicates that it partially fulfils the criterion. Additionally, the ability to capture the axial tension and bending deformation modes has not been included in the table because each formulation discussed in Section 2.5 can handle axial tension and bending.

In order to make an informed choice of beam model, one must consider the problem to be modelled. Because this thesis focuses on modelling impact in a lattice structure, the choice of beam formulation can be partially based on the requirements an impact problem imposes on the model. Most crucially, metamaterial impact problems will involve large deformations, since that is the whole point of using metamaterials as impact protection as mentioned in Subsection 2.1.4. This immediately disqualifies linear beam theories because they will not be able to capture the large deformations accurately.

This leaves the geometrically nonlinear beam formulations. Out of these, the torsion-free geometrically nonlinear Kirchhoff rod boasts the highest computational efficiency and relative to the other nonlinear theories, it is the simplest to implement. Because this formulation only depends on the primary variables $\boldsymbol{r}$, its computational cost is significantly lower than that of the Simo-Reissner beam, which requires solving for 6 unknowns. Similarly, although the general geometrically nonlinear Kirchhoff formulation and the symmetric geometrically nonlinear Kirchhoff formulation are not as computationally costly as the Simo-Reissner beam (see Subsection 2.5.4) the torsion-free formulation still has the edge over them in numerical efficiency because it does not depend on any rotational degrees of freedom.

When it comes to implementation simplicity, both the torsion-free and the symmetric geometrically nonlinear Kirchhoff formulations are significantly easier to implement than the Simo-Reissner or the general geometrically nonlinear Kirchhoff formulations because their weak forms do not depend on any reference triads. Since there is no need to parametrise additional triads, this greatly simplifies numerical procedures like spatial discretisation and linearisation. Between the torsion-free and the symmetric subcases of the geometrically nonlinear Kirchhoff formulation, one could argue that the torsion-free formulation is slightly simpler to implement because the spatial discretisation only involves discretising the primary variables $\boldsymbol{r}$ and their variation $\delta\boldsymbol{r}$ while the symmetric formulation still requires discretising the torsional degree of freedom field α and its variation $\delta\alpha$.

However, the computational efficiency gains and the relative ease of implementation of the torsion-free geometrically nonlinear Kirchhoff formulation come with some concessions and limitations as a result

of the underlying assumptions of this model. For starters, the torsion-free formulation requires the initial curvature to be zero ($\kappa_0 \equiv 0$) and the cross-section to be rotationally symmetric ($I_2 = I_3 = I_1$). It shares this limitation with the symmetric formulation. For brevity, both assumptions together will be referred to as the symmetry assumption. Neither the Simo-Reissner nor the general geometrically nonlinear Kirchhoff formulations make this assumption. However, one can make the argument that using rotationally symmetric elements with zero initial curvature are modelling choices. So long as the unit cells modelled have initially straight struts, and so long as the cross-sections of those struts are circular, then the symmetry assumption holds.

Of course, the opposite is also true: if one wants to model unit cells with curved struts for instance for conformal lattice structures [89], then one must look elsewhere than the torsion-free geometrically nonlinear Kirchhoff model. Similarly, if one would like to model unit cells with optimised strut cross-sections like using I or H sections to better resist bending, then the general geometrically nonlinear Kirchhoff formulation or the Simo-Reissner formulation are better choices. Still, for the purposes of this thesis, these two limitations are acceptable because the modelled lattice members are straight and their cross-sections are circular. The upshot in implementation simplicity due to there not being a need to parametrise additional triads is well worth it, but it pays to be mindful of the limitations imposed by the symmetry assumption.

The torsion-free geometrically nonlinear Kirchhoff model also makes assumptions on the deformation modes. The first of these assumptions is the Kirchhoff constraint (see Equation 2.33), which this formulation shares with all Kirchhoff rod formulations. The Kirchhoff constraint is based on the assumption of vanishing shear strains. As a result, all nonlinear theories in Table 3.1 except for the Simo-Reissner theory are shear-free. If the lattice structure is modelled with highly slender struts, then a shear-free formulation will suffice, which simplifies the problem because fewer unknowns must be solved. This means that if the relative density $\bar{\rho}$ is not too high, then the struts are slender enough to be modelled with shear-free elements. However, if the relative density is too high so that the struts are no longer slender, then none of the Kirchhoff formulations are valid.

The second assumption of the torsion-free geometrically nonlinear Kirchhoff theory is a characteristic of the torsion-free formulation and it imposes restrictions on the torsion deformation modes (see Equation 2.50). As a result of this, the torsion is zero along the length of the beam. Because of this, the torsional degrees of freedom can be resigned and consequently, the torsion-free formulation only depends on $\boldsymbol{r}$. This assumption makes the problem simpler and it increases the computational efficiency of the model. According to Meier et al. (2015), the torsion-free case applies to structures with finite bending stiffness but negligible torsional stiffness ($GI_T \approx 0$) [80]. This means the struts of a lattice must be very slender for the torsion-free assumption to hold, imposing an upper limit on the relative density similar to the shear-free assumption.

In all, the beam formulation of choice is the torsion-free geometrically nonlinear Kirchhoff rod. It can accurately model large deformations, and it is the most computationally efficient and the simplest of the nonlinear beam formulations. That said, it comes with limitations in terms of deformation modes and geometry.

Firstly, the torsion-free and the shear-free assumptions which underpin this beam model require the structural members to be highly slender. For lattice structures, this means that the relative density cannot be too high lest the struts become too thick. For the purposes of this thesis, this limitation can be coped with by imposing an upper limit on the relative density $\bar{\rho}$ of the modelled lattices, like $\bar{\rho} \leq 5\%$. This does mean that modelling lattices with a higher relative density requires extending the beam formulation to be valid for less slender structures. Effectively, this would entail implementing either the Simo-Reissner formulation or some other shear-rigid nonlinear beam theory. Extending the beam formulation to non-slender members would make a natural next step for any continuation on this work.

Secondly, the geometric limitations of the torsion-free geometrically nonlinear Kirchhoff model in the form of the symmetry assumption means all struts must be initially straight and must have circular cross-sections. In this thesis, these assumptions are fulfilled by only modelling unit cells with initially

straight and circular struts, but if one aims to model initially curved, non-circular struts then one must extend the beam formulation to cope with non-zero initial curvature and anisotropic bending. In this case, one must essentially implement the general geometrically nonlinear Kirchhoff rod model.

Now that the element type has been chosen, the remainder of this work is carried out with the torsion-free geometrically nonlinear Kirchhoff rod.

3.1.2. The continuum formulation of the dynamic problem

Now that the torsion-free geometrically nonlinear Kirchhoff rod model has been chosen, the weak form described by Meier (2016) could be used [90] to govern the dynamic problem. Because the configurations of beam elements of this model are determined entirely by their centreline and the orientations of their cross-sections, the beams can thus be represented with a 3D curve $\boldsymbol{r}(s,t) \in \mathbb{R}^3$ where $s \in [0,l] \subset \mathbb{R}$, the arc-length parameter, and $l \in \mathbb{R}$, the initial length, describe the parametrisation of the curve in the initial configuration. t is the independent time variable. Before moving to the next step, it is worthwhile to introduce some shorthand notations for some differential operators to keep the equations more concise. The derivative with respect to s is shortened to $(.)' = \frac{\partial}{\partial s}(.) = (.)_{,s}$. The derivative with respect to t is shortened to $\dot{(.)} = \frac{\partial}{\partial t}(.) = (.)_{,t}$.

Now, the kinetic and internal energies of a torsion-free geometrically nonlinear Kirchhoff beam can be treated. For slender torsion-free geometrically nonlinear Kirchhoff beams, it is a reasonable assumption to neglect rotational inertia contributions [90]. Furthermore, a hyperelastic constitutive law is considered. As such, the kinetic and hyperelastic stored energies per unit length are given as [90]:

$$\begin{aligned} \tilde{\Pi}_{kin} &= \frac{1}{2}\rho A \dot{\boldsymbol{r}}^T \dot{\boldsymbol{r}} \\ \tilde{\Pi}_{int} &= \frac{1}{2}EA\epsilon^2 + \frac{1}{2}EI\boldsymbol{\kappa}^T\boldsymbol{\kappa} \end{aligned} \tag{3.1}$$

Here, ρ is the reference mass density in $[kg/m^3]$, A is the reference cross-sectional area in $[m^2]$, I is the reference area moment of inertia in $[m^4]$ and E is Young's modulus in $[Pa]$. $\dot{\boldsymbol{r}}$ is the velocity field of the centreline with units $[m/s]$, ϵ ($[m/m]$) is the axial strain field, and κ ($[1/m]$) is the bending curvature field.

Integrating the expressions in Equation 3.1 over the reference length of the curve l results in Equation 3.2.

$$\begin{aligned} \Pi_{kin} &= \int_0^l \frac{1}{2}\rho A \dot{\boldsymbol{r}}^T \dot{\boldsymbol{r}} \mathrm{d}s \\ \Pi_{int} &= \int_0^l \left[\frac{1}{2}EA\epsilon^2 + \frac{1}{2}EI\boldsymbol{\kappa}^T\boldsymbol{\kappa}\right] \mathrm{d}s \end{aligned} \tag{3.2}$$

The ϵ and κ fields can be expressed in terms of $\boldsymbol{r}$ following [90] and [80]:

$$\epsilon = ||\boldsymbol{r}'|| - 1 \tag{3.3a}$$

$$\boldsymbol{\kappa} = \frac{\boldsymbol{r}' \times \boldsymbol{r}''}{||\boldsymbol{r}'||^2} \tag{3.3b}$$

$$\kappa = ||\boldsymbol{\kappa}|| \tag{3.3c}$$

Now, the terms defined in Equation 3.2 can be used to derive the weak form of the dynamic balance equations. Meier presents the weak form as such [90]:

$$\int_0^l \left[\delta\epsilon EA\epsilon + \delta\boldsymbol{\kappa}^T EI\boldsymbol{\kappa} + \delta\boldsymbol{r}^T\rho A\ddot{\boldsymbol{r}}\right] \mathrm{d}s - \int_0^l \left[\delta\boldsymbol{r}^T\tilde{\boldsymbol{f}} + \delta\boldsymbol{\theta}_\perp^T\tilde{\boldsymbol{m}}_\perp\right] \mathrm{d}s - \left[\delta\boldsymbol{r}^T\bar{\boldsymbol{f}} + \delta\boldsymbol{\theta}_\perp^T\bar{\boldsymbol{m}}_\perp\right]_{\Gamma_\sigma} \tag{3.4}$$

In Equation 3.4, $\tilde{(.)}$ denotes distributed quantities per unit reference arc-length while $\bar{(.)}$ denotes discrete quantities on the Neumann boundary Γ_σ. Also, one may notice that exclusively the moment components perpendicular to the centreline are included in the weak form. This is due to the constraint the

torsion-free assumption imposes on the moment field [90]. Finally, $\boldsymbol{\theta}_\perp$ is associated with the rotations, but formally, it represents the (perpendicular components of) the spin vector [90].

The variations present in Equation 3.4 expand to [90]:

$$\delta\epsilon = \frac{\delta \boldsymbol{r}'^T \boldsymbol{r}'}{||\boldsymbol{r}'||} \tag{3.5a}$$

$$\delta\boldsymbol{\kappa} = \frac{||\boldsymbol{r}'||^2 \left(\delta\boldsymbol{r}' \times \boldsymbol{r}'' + \boldsymbol{r}' \times \delta\boldsymbol{r}''\right) - 2\left(\delta\boldsymbol{r}'^T \boldsymbol{r}'\right)\left(\boldsymbol{r}' \times \boldsymbol{r}''\right)}{||\boldsymbol{r}'||^4} \tag{3.5b}$$

$$\delta\boldsymbol{\theta}_\perp = \frac{\boldsymbol{r}' \times \delta\boldsymbol{r}'}{||\boldsymbol{r}'||^2} \tag{3.5c}$$

With this, the weak form of the beam dynamics problem has been formulated in continuous space and time. However, in order to facilitate the introduction of fracture behaviour into the weak form in Subsection 3.1.3, it is actually more convenient to start from the strong form of the dynamic problem. For this, the approach by Kota et al. (2024) can be used [16].

The first step is integrating the balance of linear and angular momentum equations from 3D continuum theory over the cross-section of the beam (see also Subsection 2.5.3). This gives the following balance equations:

$$\boldsymbol{f}' + \tilde{\boldsymbol{f}} = \rho A \ddot{\boldsymbol{r}} \tag{3.6}$$

$$\boldsymbol{m}' + \boldsymbol{r}' \times \boldsymbol{f} + \tilde{\boldsymbol{m}} = \boldsymbol{I}_\rho \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}_\rho \boldsymbol{\omega}) \tag{3.7}$$

Some of the quantities in Equation 3.6 and Equation 3.7 have been explained before with Equation 3.1 and Equation 3.4. $\boldsymbol{I}_\rho$ ($[kg\ m^2]$) is the rotational inertia tensor, $\boldsymbol{\omega}$ ($[rad/s]$) is the axial vector belonging to the the skew-symmetric angular velocity tensor $\dot{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}^T$. $\boldsymbol{f}(s,t)$ and $\boldsymbol{m}(s,t)$ are the internal force and moment stress resultants, respectively. They are defined as follows [16]:

$$\boldsymbol{f} = \int_A \boldsymbol{P}\boldsymbol{E_1}\, \mathrm{d}A \tag{3.8}$$

$$\boldsymbol{m} = \int_A (\boldsymbol{x} - \boldsymbol{r}) \times \boldsymbol{P}\boldsymbol{E_1}\, \mathrm{d}A \tag{3.9}$$

In the above equations, $\boldsymbol{P}$ ($[Pa]$) is the Piola-Kirchhoff stress tensor, and $\boldsymbol{E_1}$ is a fixed basis vector of the intrinsic frame in the reference configuration [16].

The forces $\boldsymbol{f}$, $\tilde{\boldsymbol{f}}$, and the moments $\boldsymbol{m}$, $\tilde{\boldsymbol{m}}$ can be additively decomposed into parallel and perpendicular components. The parallel force components $\boldsymbol{f}_{||}$, $\tilde{\boldsymbol{f}}_{||}$ are the axial forces, while the perpendicular force components $\boldsymbol{f}_\perp$, $\tilde{\boldsymbol{f}}_\perp$ are the shear forces. Analogously, the parallel moment components $\boldsymbol{m}_{||}$, $\tilde{\boldsymbol{m}}_{||}$ are the torsional moments, and the perpendicular moment components $\boldsymbol{m}_\perp$, $\tilde{\boldsymbol{m}}_\perp$ are the bending moments. Here, the notation in Equation 3.10 has been used.

$$\boldsymbol{a}_{||} := \left(\boldsymbol{a} \cdot \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||}\right) \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||}, \qquad \boldsymbol{a}_\perp := \boldsymbol{a} - \boldsymbol{a}_{||} \tag{3.10}$$

As mentioned earlier in this subsection, the rotational inertia can be neglected, which is tantamount to neglecting the right-hand side of Equation 3.7. This allows for expressing the internal shear forces as a function of the bending moments after performing the cross-product of Equation 3.7 with $\boldsymbol{r}'$. The resulting equation is Equation 3.11.

$$\boldsymbol{f}_\perp = \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||^2} \times (\boldsymbol{m}'_\perp + \tilde{\boldsymbol{m}}_\perp) \tag{3.11}$$

Now, Equation 3.11 can be substituted into Equation 3.6, which gives Equation 3.12.

$$f_{||}' + \left[\frac{r'}{||r'||^2} \times (m_{\perp}' + \tilde{m}_{\perp}) \right]' + \tilde{f} = \rho A \ddot{r} \tag{3.12}$$

This equation is equivalent to the original system of governing equations Equation 3.6 and Equation 3.7, if the torsion-free formulation holds. In that case, Equation 3.11 is equivalent to Equation 3.7.

3.1.3. Introducing fracture behaviour into the beam mechanics

Before jumping into the specific spatial discretisation of the continuum formulation of the governing equation from Subsection 3.1.2, it is actually a fitting time to describe the fracture model used in the numerical framework. The reason being that the fracture behaviour of beam elements can be directly included in the weak form as Kota et al. (2024) show [16].

The discontinuous Galerkin weak form

The first step towards formulating a weak form that describes both bulk and interface material behaviour is deriving the discontinuous Galerkin formulation of Equation 3.4. To do so, consider the spatial discretisation Ω_h of a straight and undeformed beam Ω which is divided into segments $\Omega_e = (s_0^e, s_1^e)$ with $e = 1, ..., E$, so that $\Omega_h = \bigcup_{e=1}^{E} \overline{\Omega}_e$. This discretisation can also be seen in Figure 3.1. e is the element, E is the total number of elements, and s_0^e and s_1^e represent the element nodes. Furthermore, the boundary of the beam is defined as $\partial\Omega_h = \{0, L\}$, with the external normal n_e set to $n_e = 1$ at $s = L$ and $n_e = -1$ at $s = 0$. Lastly, the Neumann boundary conditions on $\partial\Omega_h$ are defined as $\partial_{N_f}\Omega_h$ and $\partial_{N_m}\Omega_h$, where the external forces $\bar{f}$ and external moments $\bar{m}_{\perp}$, respectively, are applied.

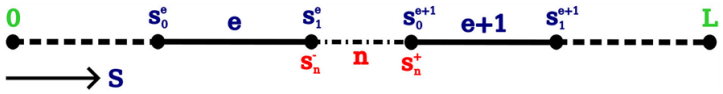


Figure 3.1: The discontinuous Galerkin discretisation of a straight and undeformed beam, from [16]. Internal nodes such as s_1^e and s_0^{e+1} are duplicated to capture discontinuities at element interfaces denoted by s_n with n the number of interfaces.

Next, Equation 3.12 is treated. In accordance with the discontinuous Galerkin formulation, the centroid position field $\boldsymbol{r}(s)$, its derivatives and its associated kinematically admissible variations $\boldsymbol{\delta r}(s)$ are allowed to exhibit discontinuities at the interfaces s_n with $n = 1, ..., E-1$ of adjacent elements. The discontinuous Galerkin weak form of Equation 3.12 can then be derived by multiplying the strong form by $\boldsymbol{\delta r}$, integrating over the individual subdomains Ω_e with $e = 1, ..., E$, and integrating that by parts. This results in Equation 3.13.

$$\begin{aligned} -\sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{f}_{||} \cdot \boldsymbol{\delta r}' \, ds - \sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{f}_{\perp} \cdot \boldsymbol{\delta r}' \, ds - \sum_{n=1}^{E-1} [\![\boldsymbol{f} \cdot \boldsymbol{\delta r}]\!] \Big|_{s_n} + (\bar{\boldsymbol{f}} \cdot \boldsymbol{\delta r}) \Big|_{\partial_{N_f}\Omega_h} \\ + \sum_{e=1}^{E} \int_{\Omega_e} \tilde{\boldsymbol{f}} \cdot \boldsymbol{\delta r} \, ds = \sum_{e=1}^{E} \int_{\Omega_e} \rho A \ddot{\boldsymbol{r}} \cdot \boldsymbol{\delta r} \, ds \end{aligned} \tag{3.13}$$

The Neumann boundary condition $\bar{f} = n_e f$ has also been applied on $\partial_{N_f}\Omega_h$ in Equation 3.13. Furthermore, the effect of the discontinuous fields has also been included as a jump term, along with the notation $\bullet\big|_a$, which means a quantity $\bullet$ evaluated at $s = a$.

The jump term can be reformulated with the identity $[\![\boldsymbol{a} \cdot \boldsymbol{b}]\!] = \langle \boldsymbol{a} \rangle \cdot [\![\boldsymbol{b}]\!] + [\![\boldsymbol{a}]\!] \cdot \langle \boldsymbol{b} \rangle$ as Equation 3.14.

$$\sum_{n=1}^{E-1} [\![\boldsymbol{f} \cdot \boldsymbol{\delta r}]\!] \Big|_{s_n} = \sum_{n=1}^{E-1} \left[\langle \boldsymbol{f} \rangle \cdot [\![\boldsymbol{\delta r}]\!] \right] \Big|_{s_n} + \sum_{n=1}^{E-1} \left[[\![\boldsymbol{f}]\!] \cdot \langle \boldsymbol{\delta r} \rangle \right] \Big|_{s_n} \tag{3.14}$$

On top of this, when the vector identity $\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b})$ is used on Equation 3.11, then the term in Equation 3.13 that includes $\boldsymbol{f}_\perp$ can be re-expressed as:

$$\begin{aligned} -\sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{f}_\perp \cdot \boldsymbol{\delta r}' \, ds &= \sum_{e=1}^{E} \int_{\Omega_e} (\boldsymbol{m}'_\perp + \tilde{\boldsymbol{m}}_\perp) \cdot \frac{\boldsymbol{r}' \times \boldsymbol{\delta r}'}{||\boldsymbol{r}'||^2} \, ds \\ &= \sum_{e=1}^{E} \int_{\Omega_e} (\boldsymbol{m}'_\perp + \tilde{\boldsymbol{m}}_\perp) \cdot \boldsymbol{\delta \theta}_\perp \, ds \end{aligned} \tag{3.15}$$

The kinematically admissible variation $\boldsymbol{\delta \theta}_\perp$ seen in Equation 3.15 and defined by Equation 2.55 emerges in the process of applying the aforementioned vector identity as the work conjugate of the bending moments. To streamline the notation in the expressions below, the subscript $\perp$ of $\boldsymbol{\delta \theta}_\perp$ is suppressed, and $\boldsymbol{\delta \theta}$ is adopted instead. Now, integrating the right-hand side of Equation 3.15 by parts results in Equation 3.16.

$$\begin{aligned} \sum_{e=1}^{E} \int_{\Omega_e} (\boldsymbol{m}'_\perp + \tilde{\boldsymbol{m}}_\perp) \cdot \boldsymbol{\delta \theta} \, ds &= \sum_{e=1}^{E} \int_{\Omega_e} \tilde{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta \theta} \, ds - \sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{m}_\perp \cdot \boldsymbol{\delta \theta}' \, ds \\ &- \sum_{n=1}^{E-1} \Big[\langle \boldsymbol{m}_\perp \rangle \cdot [\![\boldsymbol{\delta \theta}]\!] \Big] \Big|_{s_n} - \sum_{n=1}^{E-1} \Big[[\![\boldsymbol{m}_\perp]\!] \cdot \langle \boldsymbol{\delta \theta} \rangle \Big] \Big|_{s_n} + (\bar{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta \theta}) \Big|_{\partial_{N_m} \Omega_h} \end{aligned} \tag{3.16}$$

Here, the Neumann boundary condition $\bar{\boldsymbol{m}}_\perp = n_e \boldsymbol{m}_\perp$ is applied on $\partial_{N_m} \Omega_h$, and the identity $[\![\boldsymbol{a} \cdot \boldsymbol{b}]\!] = \langle \boldsymbol{a} \rangle \cdot [\![\boldsymbol{b}]\!] + [\![\boldsymbol{a}]\!] \cdot \langle \boldsymbol{b} \rangle$ was used once more.

With Equation 3.14, Equation 3.15 and Equation 3.16, Equation 3.13 can be expanded to:

$$\begin{aligned} &\sum_{e=1}^{E} \int_{\Omega_e} \rho A \ddot{\boldsymbol{r}} \cdot \boldsymbol{\delta r} \, ds + \sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{f}_{||} \cdot \boldsymbol{\delta r}' \, ds + \sum_{n=1}^{E-1} \Big[\langle f \rangle \cdot [\![\boldsymbol{\delta r}]\!] \Big] \Big|_{s_n} + \sum_{n=1}^{E-1} \Big[[\![f]\!] \cdot \langle \boldsymbol{\delta r} \rangle \Big] \Big|_{s_n} \\ &+ \sum_{e=1}^{E} \int_{\Omega_e} \boldsymbol{m}_\perp \cdot \boldsymbol{\delta \theta}' \, ds + \sum_{n=1}^{E-1} \Big[\langle \boldsymbol{m}_\perp \rangle \cdot [\![\boldsymbol{\delta \theta}]\!] \Big] \Big|_{s_n} + \sum_{n=1}^{E-1} \Big[[\![\boldsymbol{m}_\perp]\!] \cdot \langle \boldsymbol{\delta \theta} \rangle \Big] \Big|_{s_n} \\ &= \sum_{e=1}^{E} \int_{\Omega_e} \tilde{\boldsymbol{f}} \cdot \boldsymbol{\delta r} \, ds + (\bar{\boldsymbol{f}} \cdot \boldsymbol{\delta r}) \Big|_{\partial_{N_f} \Omega_h} + \sum_{e=1}^{E} \int_{\Omega_e} \tilde{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta \theta} \, ds + (\bar{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta \theta}) \Big|_{\partial_{N_m} \Omega_h} \end{aligned} \tag{3.17}$$

The expression has been shuffled a bit so that the external load terms and the Neumann boundary terms are on one side of the equality sign, and the rest on the other side.

Because force $\boldsymbol{f}$ and bending moment $\boldsymbol{m}_\perp$ discontinuities are not required to be penalised to preserve the consistency of the numerical formulation, the associated terms in Equation 3.17 may be disregarded [91] [92]. Still, numerical stability requires inter-element compatibility to be enforced weakly. This is accomplished with the interior penalty method as proposed in [91] [92]. In this case, the terms below are incorporated into the left-hand side of Equation 3.17:

$$\sum_{n=1}^{E-1} \beta_p \left[\left\langle \frac{EA}{h} \right\rangle [\![\boldsymbol{\delta r}]\!] \cdot [\![\boldsymbol{r}]\!] \right] \Big|_{s_n} + \sum_{n=1}^{E-1} \beta_t \left[\left\langle \frac{EI}{h} \right\rangle [\![\boldsymbol{\delta g_1}]\!] \cdot [\![\boldsymbol{g_1}]\!] \right] \Big|_{s_n} \tag{3.18}$$

Here, $\boldsymbol{g_1} = \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||}$ due to the Kirchhoff constraint (see Equation 2.33). $\beta_p > 1$ ([−]) and $\beta_t > 1$ ([−]) are the position and tangent jump penalty parameters, respectively. Lastly, h is the element size in $[m]$.

With this in place, and after substituting the constitutive relations for torsion-free geometrically nonlinear Kirchhoff beams given by Equation 2.54, the stabilised discontinuous Galerkin weak form can be expressed as such:

$$\begin{aligned}
&\int_{\Omega_h} \rho A \ddot{\boldsymbol{r}} \cdot \boldsymbol{\delta r}\, ds + \int_{\Omega_h} EA\, \varepsilon\, \delta\varepsilon\, ds + \sum_{n=1}^{E-1} \left[\langle \boldsymbol{f} \rangle \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} \beta_p \left[\left\langle \frac{EA}{h} \right\rangle [\![\boldsymbol{r}]\!] \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} \\
&\qquad + \int_{\Omega_h} EI\, \boldsymbol{\kappa} \cdot \boldsymbol{\delta\kappa}\, ds + \sum_{n=1}^{E-1} \left[\langle \boldsymbol{m_\perp} \rangle \cdot [\![\boldsymbol{\delta\theta}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} \beta_t \left[\left\langle \frac{EI}{h} \right\rangle [\![\boldsymbol{g_1}]\!] \cdot [\![\boldsymbol{\delta g_1}]\!]\right]\Big|_{s_n} \\
&= \int_{\Omega_h} \tilde{\boldsymbol{f}} \cdot \boldsymbol{\delta r}\, ds + \left(\bar{\boldsymbol{f}} \cdot \boldsymbol{\delta r}\right)\Big|_{\partial_{N_f}\Omega_h} + \int_{\Omega_h} \tilde{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta\theta}\, ds + \left(\bar{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta\theta}\right)\Big|_{\partial_{N_m}\Omega_h}
\end{aligned} \tag{3.19}$$

Incorporating the cohesive zone model into the discontinuous Galerkin weak form
Everything in this subsection has been building up to this: the inclusion of fracture behaviour directly into the weak form. This is why the first half of this subsection described the derivation of the discontinuous Galerkin weak form, because that formulation of the weak form presents an opportunity for tying interface and bulk behaviour in an elegant way. Kota et al. (2024) show how the cohesive zone model (see Subsection 2.7.3) can be integrated seamlessly into the weak form [16].

It has been mentioned earlier that $\boldsymbol{r}(s)$ and $\boldsymbol{r'}(s)$ may be discontinuous at the interfaces of the finite elements. That said, before fracture occurs, the compatibility across element boundaries is ensured through the variationally consistent interface forces and moments. These are the two jump terms in Equation 3.19 that involve the average force $\langle \boldsymbol{f} \rangle$ and average bending moment $\langle \boldsymbol{m_\perp} \rangle$ fields. For brevity, they will be referred to as the discontinuous Galerkin (DG) forces and moments.

When the fracture criterion (Equation 2.127) is met, the axial interface forces $\langle \boldsymbol{f} \rangle_{||}$ and the interface bending moments $\langle \boldsymbol{m}_\perp \rangle$ in Equation 3.19 are replaced with the cohesive axial forces and bending moments of Equation 2.124. Only the axial component of the interface forces are replaced; not the full interface forces, because Kirchhoff beams assume negligible shear strains, and in accordance with that assumption, no shear fracture modes are modelled. This means that the component of the interface force field $\langle \boldsymbol{f} \rangle$ in Equation 3.19 orthogonal to the cohesive boundary's unit normal $\hat{\boldsymbol{n}}_{\boldsymbol{coh}}$ is unchanged until complete interface failure [92], when this term finally deactivated.

This switching from DG interface loads to cohesive loads is controlled by two binary parameters α_n and γ_n at each interface n. Prior to fracture initiation at interface n, $\alpha_n = 1$ and $\gamma_n = 1$. After the fracture criterion Equation 2.127 is satisfied, the parameter α_n is set to $\alpha_n = 0$, but γ_n still maintains the value 1. It is only after complete decohesion, when $\Delta \geq \Delta_c$, that γ_n also takes the value 0. Introducing the two switching parameters into Equation 3.19 results in Equation 3.20.

$$\begin{aligned}
&\int_{\Omega_h} \rho A \ddot{\boldsymbol{r}} \cdot \boldsymbol{\delta r}\, ds + \int_{\Omega_h} EA\, \varepsilon\, \delta\varepsilon\, ds + \sum_{n=1}^{E-1} \alpha_n \left[\boldsymbol{f_{DG,||}} \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} \gamma_n \left[\boldsymbol{f_{DG,\perp}} \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} \\
&\qquad + \sum_{n=1}^{E-1} \alpha_n \left[\beta_p \left\langle \frac{EA}{h} \right\rangle \boldsymbol{c_{DG,||}} \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} \gamma_n \left[\beta_p \left\langle \frac{EA}{h} \right\rangle \boldsymbol{c_{DG,\perp}} \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} \\
&\qquad + \int_{\Omega_h} EI\, \boldsymbol{\kappa} \cdot \boldsymbol{\delta\kappa}\, ds + \sum_{n=1}^{E-1} \alpha_n \left[\langle \boldsymbol{m_\perp} \rangle \cdot [\![\boldsymbol{\delta\theta}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} \alpha_n \left[\beta_t \left\langle \frac{EI}{h} \right\rangle [\![\boldsymbol{g_1}]\!] \cdot [\![\boldsymbol{\delta g_1}]\!]\right]\Big|_{s_n} \\
&\qquad + \sum_{n=1}^{E-1} (1-\alpha_n) \left[\boldsymbol{f_{coh,||}} \cdot [\![\boldsymbol{\delta r}]\!]\right]\Big|_{s_n} + \sum_{n=1}^{E-1} (1-\alpha_n) \left[\boldsymbol{m_{coh,\perp}} \cdot [\![\boldsymbol{\delta g_1}]\!]\right]\Big|_{s_n} \\
&= \int_{\Omega_h} \tilde{\boldsymbol{f}} \cdot \boldsymbol{\delta r}\, ds + \left(\bar{\boldsymbol{f}} \cdot \boldsymbol{\delta r}\right)\Big|_{\partial_{N_f}\Omega_h} + \int_{\Omega_h} \tilde{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta\theta}\, ds + \left(\bar{\boldsymbol{m}}_\perp \cdot \boldsymbol{\delta\theta}\right)\Big|_{\partial_{N_m}\Omega_h}
\end{aligned} \tag{3.20}$$

Above, the following shorthand notations are used:

$$
\begin{aligned}
\boldsymbol{f}_{DG,||} &= \left(\langle \boldsymbol{f} \rangle \cdot \hat{\boldsymbol{n}}_{coh}\right) \hat{\boldsymbol{n}}_{coh} \\
\boldsymbol{f}_{DG,\perp} &= (\boldsymbol{I} - \hat{\boldsymbol{n}}_{coh} \otimes \hat{\boldsymbol{n}}_{coh}) \langle \boldsymbol{f} \rangle \\
\boldsymbol{c}_{DG,||} &= \left(\llbracket \boldsymbol{r} \rrbracket \cdot \hat{\boldsymbol{n}}_{coh}\right) \hat{\boldsymbol{n}}_{coh} \\
\boldsymbol{c}_{DG,\perp} &= (\boldsymbol{I} - \hat{\boldsymbol{n}}_{coh} \otimes \hat{\boldsymbol{n}}_{coh}) \llbracket \boldsymbol{r} \rrbracket
\end{aligned}
$$

In Equation 3.20, one can see that the two binary parameters α_n and γ_n are used to switch between interface behaviour: before crack initiation ($\alpha_n = 1$ and $\gamma_n = 1$), the interface loads are the DG forces and moments; during crack growth ($\alpha_n = 0$ and $\gamma_n = 1$), the axial interface forces and bending moments belong to the cohesive zone model but the transverse interface forces are unchanged; after decohesion ($\alpha_n = 0$ and $\gamma_n = 0$), the transverse interface forces are also deactivitated. Furthermore, if the cracked surfaces come into contact again after fracture onset, that is when $\Delta_{||} < 0$ and $\Delta_{max} > 0$, then the DG axial forces and the stabilisation term associated with the position jump $\llbracket \boldsymbol{\delta r} \rrbracket$ are reactivated in order to let compressive stress waves propagate across the cracked interfaces.

Equation 3.20 is the weak form used by the numerical framework synthesised in this thesis, but it is by no means the only way to integrate interface behaviour with bulk behaviour. However, the fact that the derivation by Kota et al. (2024) seamlessly ties bulk and fracture behaviour together, and the fact that Equation 3.20 is already implemented in SUMMIT, makes it an easy choice for settling on this formulation.

3.1.4. Spatial and temporal discretisation

With the integration of fracture behaviour out of the way, now the spatial and temporal discretisation of the weak form can be treated. Covering the spatial discretisation before the contact model is more convenient since the contact formulation explicitly makes use of the fact that the beams are parametrised. The contact forces are then incorporated into Equation 3.20 as explained in Subsection 3.1.7. The following will mainly be formulated with the intention of solving Equation 3.20 numerically, however, the discretisation applies to Equation 3.4 or any other formulation of the weak form just as much, if one is not as inclined to follow the method described by Kota et al. (2024).

Discretisation in space

First, $\boldsymbol{r}$ is discretised in space using third-order Hermite polynomials. It has been shown that these basis functions can reach an order of convergence of four in continuous Galerkin settings [69]. Hermite polynomials are preferred over Lagrange polynomials because when Hermite polynomials are used, the arc-length derivative of $\boldsymbol{r}$, $\boldsymbol{r'}$, at each node is directly accessible as a primary nodal degree of freedom. This perk makes the computation of the terms at the element interfaces in Equation 3.20 simpler. As such, the discretisation of $\boldsymbol{r}$ is the following:

$$
\boldsymbol{r}(\xi) \approx \sum_{b=1}^{2} N_p^b(\xi)\boldsymbol{p}^b + \frac{L}{2}\sum_{b=1}^{2} N_t^b(\xi)\boldsymbol{t}^b := \sum_{a=1}^{4} N_a(\xi)\boldsymbol{x_a} \tag{3.21}
$$

In Equation 3.21, $\boldsymbol{p}^b$ and $\boldsymbol{t}^b$ signify the position and tangent degrees of freedom at the element nodes $b = 1, 2$. $\xi \in [-1, 1]$ represents the parametric coordinate on the reference element. ξ is mapped to the arc-length coordinate $s \in [0, L]$ through the linear transformation $s = \frac{L}{2}(1 + \xi)$. The two functions of ξ, N_p^b and N_t^b, are defined as the shape functions below.

$$
\begin{aligned}
N_p^1(\xi) &= \frac{1}{4}(2 + \xi)(1 - \xi)^2 \qquad N_p^2(\xi) = \frac{1}{4}(2 - \xi)(1 + \xi)^2 \\
N_t^1(\xi) &= \frac{1}{4}(1 + \xi)(1 - \xi)^2 \qquad N_t^2(\xi) = -\frac{1}{4}(1 - \xi)(1 + \xi)^2
\end{aligned} \tag{3.22}
$$

The spatial discretisation gives rise to the semi-discrete system of equations Equation 3.23.

$$
\boldsymbol{M_{ab}}\ddot{\boldsymbol{x}}_{\boldsymbol{b}} + \boldsymbol{f_a^{int}} + \boldsymbol{f_{a^{\pm}}^{jump}} = \boldsymbol{f_a^{ext}} \tag{3.23}
$$

Here, $\boldsymbol{M}_{ab}\,\ddot{\boldsymbol{x}}_b$ represents the inertial forces, $\boldsymbol{f}_a^{\text{int}}$ are the internal bulk forces, $\boldsymbol{f}_{a^\pm}^{\text{jump}}$ signifies the internal interface forces, and $\boldsymbol{f}_a^{\text{ext}}$ comprise the external forces. The exact expressions of $\boldsymbol{M}_{ab}\,\ddot{\boldsymbol{x}}_b$, $\boldsymbol{f}_a^{\text{int}}$, $\boldsymbol{f}_{a^\pm}^{\text{jump}}$ and $\boldsymbol{f}_a^{\text{ext}}$ in terms of $\boldsymbol{r}$, the shape functions, the constitutive parameters and the geometry can be found in Appendix A of Kota et al. (2024) [16].

Discretisation in time
Because of the severe nonlinearities which will rear their heads when modelling impact and fracture, an explicit time marching scheme is used. Not only is an explicit scheme simpler than an implicit scheme, but explicit time schemes are generally the time integration schemes of choice when modelling wave propagation and impact problems [93]. In this case, Equation 3.23 is solved with a second-order explicit Newmark scheme. Special mass lumping [94] is used to prevent having to solve a linear system to calculate the nodal accelerations, thereby saving on computational costs. However, explicit time marching schemes are conditionally stable, so the time step Δt must be small enough to prevent numerical instability. One way of choosing the time step size is by computing the stable time step size Δt_c according to the Courant-Friedrichs-Lewy (CFL) condition Equation 3.24.

$$\Delta t_c = \frac{2}{\omega_{max}} \tag{3.24}$$

Here, $\omega_{max} = \max_{i=1}^{N}(||\lambda_i||)$ is the largest natural frequency of the system.

Using Equation 3.24, the stable time step size Δt_c can be calculated one time before the simulation, and over the duration of the simulation, the time step can simply be held constant and equal to the stable time step, that is $\Delta t = \Delta t_c$. To actually calculate Δt_c, one must solve the linearised eigenvalue problem below.

$$\left(\boldsymbol{K}_{ab}^{\text{int}} + \boldsymbol{K}_{ab^\pm}^{\text{jump,DG}} - \lambda^2\,\boldsymbol{M}_{ab}^{\text{lump}}\right)\boldsymbol{\Phi} = \boldsymbol{0} \tag{3.25}$$

In Equation 3.25, λ is an eigenvalue, and $\boldsymbol{\Phi}$ is its corresponding eigenvector. $\boldsymbol{M}_{ab}^{\text{lump}}$ is the lumped mass matrix of the system, and $\boldsymbol{K}_{ab}^{\text{int}}$ and $\boldsymbol{K}_{ab^\pm}^{\text{jump,DG}}$ are stiffness matrices. The exact expressions for $\boldsymbol{K}_{ab}^{\text{int}}$ and $\boldsymbol{K}_{ab^\pm}^{\text{jump,DG}}$ can be found in Appendix B of Kota et al. (2024) [16].

The specific second order Newmark scheme used to solve Equation 3.23 is a central difference scheme. However, to better reflect the solver implementation in SUMMIT, Equation 3.23 will be reformulated in residual form.

$$\boldsymbol{M}\ddot{\boldsymbol{x}} = \boldsymbol{R}(\boldsymbol{x}) = -\boldsymbol{f}^{\boldsymbol{int}} - \boldsymbol{f}_{\boldsymbol{\pm}}^{\boldsymbol{jump}} + \boldsymbol{f}^{\boldsymbol{ext}} \tag{3.26}$$

Here, $\boldsymbol{R}$ is the residual force vector, which, in the case of the central difference scheme, is a function of the nodal degrees of freedom, $\boldsymbol{x}$.

The update equation for the nodal degrees of freedom, $\boldsymbol{x}$ is given below.

$$\Delta\boldsymbol{x}^{n+1} = \Delta t\,\dot{\boldsymbol{x}}^n + \frac{\Delta t^2}{2}\ddot{\boldsymbol{x}}^n \tag{3.27}$$

In Equation 3.27, $\dot{\boldsymbol{x}}^n$ is the nodal velocity field at time t_n, $\ddot{\boldsymbol{x}}^n$ is the nodal acceleration field at time t_n, and $\Delta\boldsymbol{x}^{n+1}$ is the nodal degrees of freedom increment for time t_{n+1}.

After predicting the nodal degrees of freedom at time t_{n+1}, $\boldsymbol{x}^{n+1}$, the nodal acceleration field at time t_{n+1} can be computed by solving Equation 3.28 explicitly for $\ddot{\boldsymbol{x}}^{n+1}$.

$$\boldsymbol{M}^{\text{lump}}\ddot{\boldsymbol{x}}^{n+1} = \boldsymbol{R}(\boldsymbol{x}^{n+1}) \tag{3.28}$$

Important to note is that the mass matrix in Equation 3.28, $\boldsymbol{M}^{\text{lump}}$, is the lumped mass matrix. In this case, the equation can be solved explicitly without needing to perform any matrix inversions or solving a linear system.

With the nodal acceleration field at time t_{n+1} known, the nodal velocity field at time t_{n+1} can also be calculated using the update equation below.

$$\dot{\boldsymbol{x}}^{n+1} = \dot{\boldsymbol{x}}^{n} + \frac{\Delta t}{2}\left(\ddot{\boldsymbol{x}}^{n} + \ddot{\boldsymbol{x}}^{n+1}\right) \tag{3.29}$$

With the kinematic update equations Equation 3.27, Equation 3.29, and the acceleration update equation Equation 3.28, the full time marching scheme has been discussed.

3.1.5. Choice of beam-to-beam contact formulation

Subsection 3.1.3 offers a way of unifying fracture behaviour with bulk material behaviour. However, still missing is the contribution of beam-to-beam contact to the equation of motion as given by Equation 3.23. Therefore, the implementation of the contact model is treated next.

The natural starting point is deciding on the formulation for beam-to-beam contact. The literature review on contact mechanics, and specifically on beam-to-beam contact (Subsection 2.6.6) concluded that a penalty contact method is the way forward, but it left off with two ways of formulating the beam contact problem: point-to-point contact and line-to-line contact. Both formulations have their strengths and weaknesses, so similar to Subsection 3.1.1, the choice is based on some criteria.

These criteria are (1) robustness, (2) simplicity and (3) computational cost. Robustness refers to the contact formulation's ability to handle arbitrary contact angles. Simplicity is the ease with which a contact model based on this formulation can be implemented in SUMMIT and integrated with the solution routine described in Subsection 3.1.4. Lastly, computational cost signifies the time it takes for the contact algorithm to find contact. Subsection 2.6.6 already treated the strengths and weaknesses of both formulations in detail, so the performance of both contact formulations have been summarised in Table 3.2.

Table 3.2: Overview of the beam-to-beam contact formulations.

Formulations	Robustness	Simplicity	Computational cost
Point-to-point	-	+	+
Line-to-line	+	-	-

Meier et al. (2016) make it quite clear that the line-to-line formulation is more robust than the point-to-point formulation, being able to model arbitrary beam contact scenarios from the full range of possible contact angles $\alpha \in [0°, 90°]$ [17]. That said, this formulation is a lot more complex to implement, and more computationally expensive than the point-contact formulation.

It is for this reason that the framework synthesised in this work relies on the point-to-point beam contact formulation. The biggest motivation is the fact that the point-contact formulation is relatively easy to implement in SUMMIT, and can be incorporated into the solver routine with minimal adjustments. The line-contact formulation by contrast requires some substantial changes to the representation of the elements since the contact discretisation is decoupled from the beam discretisation. This also changes the residual contributions of the integration points used by the contact model (see Equation 2.116). Because this thesis represents a preliminary attempt, and in the context of the SUMMIT framework, the first stage, to bridge the gap between fracture and beam contact, it was thought wise to limit to contact model to the simplest formulation possible.

Having said this, conceding the robustness offered by the line-contact formulation is not the end of the world. The point-contact formulation cannot guarantee a unique solution when contact angles are small, but such contact events usually are forms of contact between parallel members or intertwined members. Meier et al. (2016) provides some examples of such contact cases (Figure 2.40) [17]. The structures modelled in this thesis consist of beam lattice unit cells like the insideBCC (Figure 4.1), the octet (Figure 2.9) and the truncated octahedron (Figure 2.11), so unless the members are pre-wound,

it seems unlikely that they can intertwine during impact. Furthermore, the Newton solver used to find the contact point will automatically return the mid-point of the beam if it cannot find a solution, meaning that if enough beam elements are used, the real contact point may not be that far from the calculated contact point.

The computational cost criterion is saved for last because it does not really factor into choosing the beam contact formulation. As mentioned in Subsection 2.6.6 and above, the line-contact formulation introduces additional computational complexity. In response to this, Meier et al. (2017) proposed a hybrid method which combines point-contact, for large contact angles, and line-contact, for small contact angles, to essentially get the benefits of both worlds: compared to pure line-contact, it is computationally less expensive; compared to pure point-contact, it is more robust [95]. Be that as it may, the existence of this hybrid model does not change the fact that a requirement is implementing a line-contact model first. Therefore, the computational cost criterion leads back to the simplicity criterion, because between the pure point-contact model and the hybrid contact model, the pure point-contact model is still the preferable alternative for the purposes of this study because of its simplicity. This also means that if future works endeavour to implement a line-contact model, a natural evolution would then be to develop the hybrid model.

One final remark is that the point-contact formulation can be made slightly more robust than the original formulation by Wriggers and Zavarise (1997) [85] by using a phase angle coordinate transformation. This is explained in more detail in Subsection 3.1.6.

3.1.6. Phase angle coordinate transformation

The use of a phase angle coordinate transformation is inspired by Sonandres (2025) to make the contact algorithm more robust [96]. In his work, he introduces a phase angle mapping to convert the constrained minimisation problem $g = d_{bl} - (R_1 + R_2)$, into an unconstrained one. Performing the optimisation on the parametric coordinates ξ_c and η_c is problematic because ξ_c and η_c are restricted to the interval $[-1, 1]$. The standard Newton method is not inherently designed for constrained optimisation. To solve this, one could use either a penalty or a barrier method to impose the constraints, but these approaches require modifications to the standard Newton algorithm and therefore introduce additional computational complexity and cost. Alternatively, one can implement corrective steps within the solver algorithm to project the solution back into the feasible domain if the Newton increment violets the bounds. However, such an approach can degrade the convergence properties of the optimisation algorithm.

Instead, the adopted approach is introducing a phase angle mapping to transforms the search space to a continuous manifold through the following coordinate transformation:

$$\boldsymbol{r} = \boldsymbol{r}(\xi(\phi)), \quad \xi = \sin(\phi), \quad \phi \in \mathfrak{R} \tag{3.30}$$

Because the sine function is periodic, this means that any real valued ϕ maps to a unique ξ in the range $[-1, 1]$. Therefore, no explicit constraints in the optimisation are needed, since the constraints have been incorporated into the parametrisation.

Because of this coordinate transformation, Equation 2.92 must be reformulated to reflect the change in coordinates. This involves applying the chain rule (repeatedly) to all the terms in Equation 2.92 which are a function of ξ or η. Because Jake Sonandres used a slightly different notation and linearised solution system formulation, the following equations differ slightly.

First, the two coordinate transformations:

$$\xi = \sin(\phi_1) \tag{3.31}$$

$$\eta = \sin(\phi_2) \tag{3.32}$$

To help make the notation used in Equation 2.92 a bit more concise, consider that the optimisation problem attempts to find the optima of the following objective function:

$$f(\xi, \eta) = \frac{1}{2} \left[\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)\right]^T \left[\boldsymbol{r}_1(\xi) - \boldsymbol{r}_2(\eta)\right] \tag{3.33}$$

One can see that taking the first, second and mixed partial derivatives of Equation 3.33 with respect to ξ and η results in the same terms in Equation 2.92. As such, the system below is equivalent to Equation 2.92.

$$\begin{bmatrix} \frac{\partial^2 f}{\partial \xi^2} & \frac{\partial^2 f}{\partial \xi \partial \eta} \\ \frac{\partial^2 f}{\partial \eta \partial \xi} & \frac{\partial^2 f}{\partial \eta^2} \end{bmatrix} \begin{bmatrix} \Delta \xi_c \\ \Delta \eta_c \end{bmatrix} = - \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \end{bmatrix} \tag{3.34}$$

Equation 3.34 is transformed into Equation 3.35 to reflect the coordinate transformation.

$$\begin{bmatrix} \frac{\partial^2 f}{\partial \phi_1^2} & \frac{\partial^2 f}{\partial \phi_1 \partial \phi_2} \\ \frac{\partial^2 f}{\partial \phi_2 \partial \phi_1} & \frac{\partial^2 f}{\partial \phi_2^2} \end{bmatrix} \begin{bmatrix} \Delta \phi_1 \\ \Delta \phi_2 \end{bmatrix} = - \begin{bmatrix} \frac{\partial f}{\partial \phi_1} \\ \frac{\partial f}{\partial \phi_2} \end{bmatrix} \tag{3.35}$$

Because $f = f(\boldsymbol{r}_1, \boldsymbol{r}_2) = f(\boldsymbol{r}_1(\xi), \boldsymbol{r}_2(\eta)) = f(\boldsymbol{r}_1(\xi(\phi_1)), \boldsymbol{r}_2(\eta(\phi_2)))$, it means that each term in Equation 3.35 is found by using the chain rule.

First, the first partial derivatives with respect to ϕ_1 and ϕ_2 are treated.

$$\frac{\partial f}{\partial \phi_1} = \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} \tag{3.36}$$

$$\frac{\partial f}{\partial \phi_2} = \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} \tag{3.37}$$

The second order partial derivatives with respect to ϕ_1 and ϕ_2 are given below.

$$\begin{aligned} \frac{\partial^2 f}{\partial \phi_1^2} &= \frac{\partial}{\partial \phi_1} \left(\frac{\partial f}{\partial \boldsymbol{r}_1} \right) \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial}{\partial \phi_1} \left(\frac{\partial \boldsymbol{r}_1}{\partial \xi} \right) \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial^2 \xi}{\partial \phi_1^2} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_1^2} \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial^2 \boldsymbol{r}_1}{\partial \xi^2} \frac{\partial \xi}{\partial \phi_1} \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial^2 \xi}{\partial \phi_1^2} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_1^2} \frac{\partial \boldsymbol{r}_1}{\partial \xi} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \left(\frac{\partial \xi}{\partial \phi_1} \right)^2 + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \left(\frac{\partial^2 \boldsymbol{r}_1}{\partial \xi^2} \left(\frac{\partial \xi}{\partial \phi_1} \right)^2 + \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial^2 \xi}{\partial \phi_1^2} \right) \end{aligned} \tag{3.38}$$

$$\begin{aligned} \frac{\partial^2 f}{\partial \phi_2^2} &= \frac{\partial}{\partial \phi_2} \left(\frac{\partial f}{\partial \boldsymbol{r}_2} \right) \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial}{\partial \phi_2} \left(\frac{\partial \boldsymbol{r}_2}{\partial \eta} \right) \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial^2 \eta}{\partial \phi_2^2} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_2^2} \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial^2 \boldsymbol{r}_2}{\partial \eta^2} \frac{\partial \eta}{\partial \phi_2} \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial^2 \eta}{\partial \phi_2^2} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_2^2} \frac{\partial \boldsymbol{r}_2}{\partial \eta} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \left(\frac{\partial \eta}{\partial \phi_2} \right)^2 + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \left(\frac{\partial^2 \boldsymbol{r}_2}{\partial \eta^2} \left(\frac{\partial \eta}{\partial \phi_2} \right)^2 + \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial^2 \eta}{\partial \phi_2^2} \right) \end{aligned} \tag{3.39}$$

$$\begin{aligned} \frac{\partial^2 f}{\partial \phi_1 \partial \phi_2} &= \frac{\partial}{\partial \phi_1} \left(\frac{\partial f}{\partial \boldsymbol{r}_2} \right) \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial}{\partial \phi_1} \left(\frac{\partial \boldsymbol{r}_2}{\partial \eta} \right) \frac{\partial \eta}{\partial \phi_2} + \frac{\partial f}{\partial \boldsymbol{r}_2} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial^2 \eta}{\partial \phi_1 \partial \phi_2} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_1 \partial \boldsymbol{r}_2} \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} \cdot \frac{\partial \boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} \end{aligned} \tag{3.40}$$

$$\begin{aligned} \frac{\partial^2 f}{\partial \phi_2 \partial \phi_1} &= \frac{\partial}{\partial \phi_2} \left(\frac{\partial f}{\partial \boldsymbol{r}_1} \right) \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial}{\partial \phi_2} \left(\frac{\partial \boldsymbol{r}_1}{\partial \xi} \right) \frac{\partial \xi}{\partial \phi_1} + \frac{\partial f}{\partial \boldsymbol{r}_1} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial^2 \xi}{\partial \phi_2 \partial \phi_1} \\ &= \frac{\partial^2 f}{\partial \boldsymbol{r}_2 \partial \boldsymbol{r}_1} \frac{\boldsymbol{r}_2}{\partial \eta} \frac{\partial \eta}{\partial \phi_2} \cdot \frac{\partial \boldsymbol{r}_1}{\partial \xi} \frac{\partial \xi}{\partial \phi_1} \end{aligned} \tag{3.41}$$

Equation 3.40 and Equation 3.41 make use of the fact that $\boldsymbol{r}_1$ and ξ, and $\boldsymbol{r}_2$ and η are functions of ϕ_1 and ϕ_2, respectively, to immediately ignore the last two terms.

What remains, is finding the expressions for the partial derivatives with respect to $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, the partial derivatives with respect to ξ and η, and finally, the partial derivatives with respect to ϕ_1 and ϕ_2. Starting off with the partial derivatives of the objective function f with respect to $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, the first order, second order and mixed partial derivatives are listed below.

$$\frac{\partial f}{\partial \boldsymbol{r}_1} = \boldsymbol{r}_1 - \boldsymbol{r}_2, \quad \frac{\partial f}{\partial \boldsymbol{r}_2} = -(\boldsymbol{r}_1 - \boldsymbol{r}_2) \tag{3.42}$$

$$\frac{\partial^2 f}{\partial \boldsymbol{r}_1^2} = \boldsymbol{I}, \quad \frac{\partial^2 f}{\partial \boldsymbol{r}_2^2} = \boldsymbol{I} \tag{3.43}$$

$$\frac{\partial^2 f}{\partial \boldsymbol{r}_1 \partial \boldsymbol{r}_2} = -\boldsymbol{I}, \quad \frac{\partial^2 f}{\partial \boldsymbol{r}_2 \partial \boldsymbol{r}_1} = -\boldsymbol{I} \tag{3.44}$$

The partial derivatives of $\boldsymbol{r}_1$ with respect to ξ, and $\boldsymbol{r}_2$ with respect to η are actually found by taking the corresponding derivative of $\boldsymbol{r}$ from Equation 3.21. This merely means taking the derivatives of the shape functions from Equation 3.22. SUMMIT has an efficient way of storing the values of the derivatives of the shape functions, which the contact algorithm takes advantage of.

The last partial derivatives that must be addressed, are the derivatives of ξ and η with respect to ϕ_1 and ϕ_2, respectively. They are listed below.

$$\frac{\partial \xi}{\partial \phi_1} = \cos(\phi_1) \tag{3.45}$$

$$\frac{\partial^2 \xi}{\partial \phi_2^2} = -\sin(\phi_2) \tag{3.46}$$

The derivatives for η with respect to ϕ_2 are the exact same as Equation 3.45 and Equation 3.46, except ϕ_1 must be substituted with ϕ_2. With this, all terms in Equation 3.35 are now known.

3.1.7. Contact model

With the core of the contact algorithm set, that is the optimisation problem and its linearised system described in Subsection 3.1.6, it is now time to cover the inner workings of the contact model. The contact model revolves around the contact detection process followed by contact enforcement.

Contact detection routine

This routine determines whether two beams are in a state of contact, and if so, it calculates the contact force. To reiterate, a contact event occurs when the interpenetration condition (Equation 2.91) from Subsection 2.6.6, and repeated below as Equation 3.47.

$$g = d_{bl} - (R_1 + R_2) \geq 0 \tag{3.47}$$

Equation 3.47 requires a pair of elements to evaluate the condition. Therefore, the first step of contact detection is finding element pairs which may violate Equation 3.47. Collecting these element pairs is done using a linear nearest neighbour search. This subroutine is described visually in Figure 3.2.

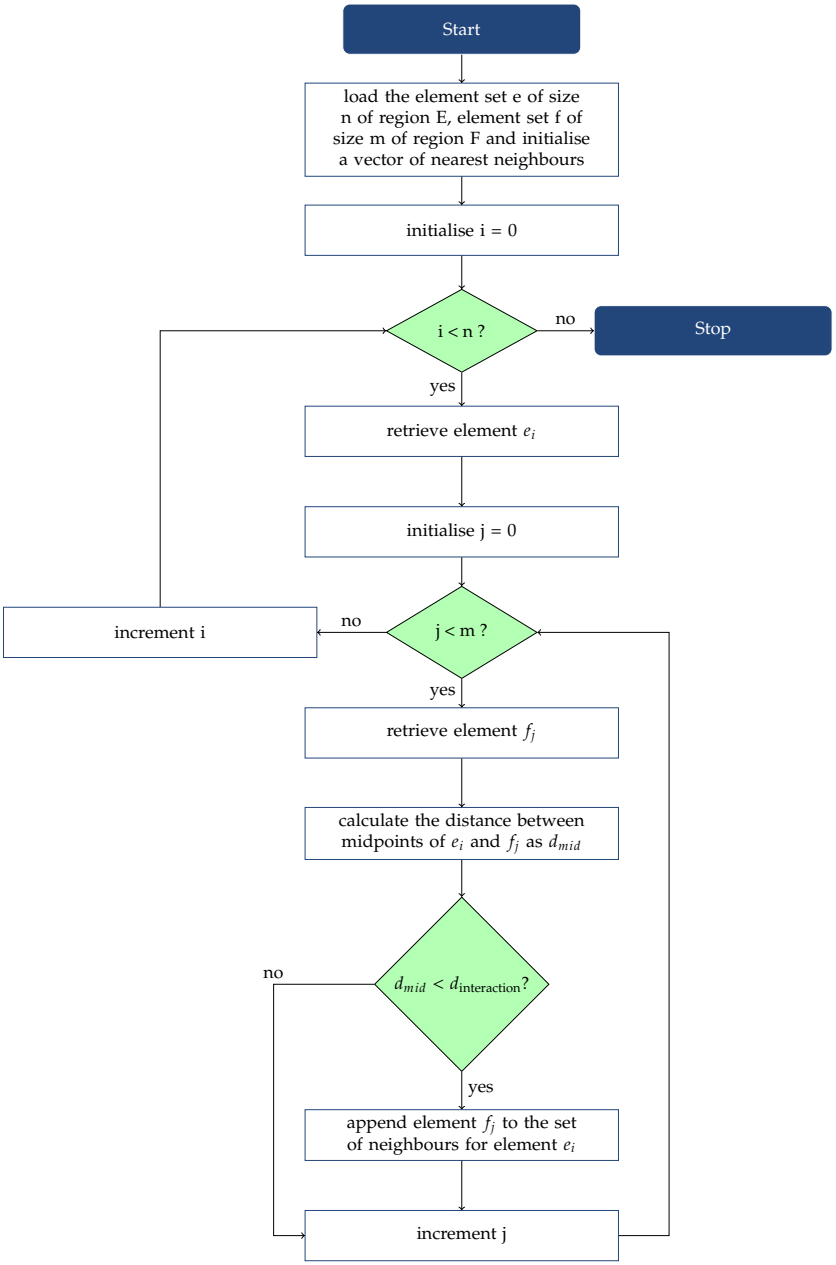


Figure 3.2: A flowchart of the nearest neighbour search subroutine.

The flow chart Figure 3.2 mentions $d_{interaction}$. This is the interaction distance between elements, and it is defined as $d_{\text{interaction}} = h_{\max} + (R_i + R_j)$. $h_{\max}$ is the maximum element length in $[m]$, while R_i and R_j are the radii of the element pair, in $[m]$.

The subroutine described in Figure 3.2 limits the search space of the contact detection algorithm to the pairs of elements which are close enough together. This way, only the element pairs which may be in contact are left. From here on, the nearest neighbour element pairs are tested against Equation 3.47. If an element pair does indeed violate the interpenetration condition, then its contact state is set to true, and the contact detection routine proceeds to calculate the contact force. The entire contact detection routine is presented visually in Figure 3.3. The entire routine outlined below is executed at every time step.

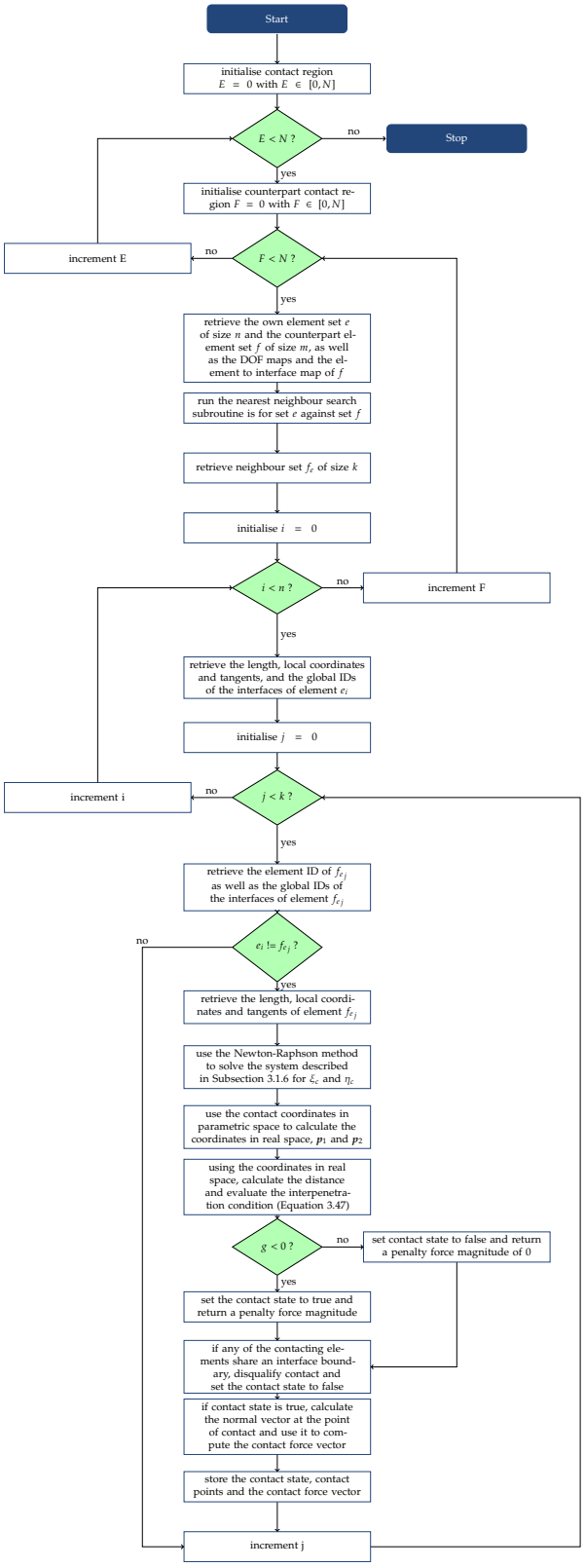


Figure 3.3: A flowchart of the contact detection routine.

One particularity of the contact detection algorithm which bears mentioning is the fact that contact between elements with a shared interface boundary is not allowed. This means that any contact event for element pairs that share an interface boundary is disregarded. The reason for this is that the nearest

neighbour search is agnostic about element connectivity: it does not care whether the element pairs are connected or not, as long as they are within the interaction distance. The result is a phenomenon where where beam elements sharing an interface experience contact when they should not. This means that the nearest neighbour search will always find element pairs which are simply two elements connected at one of their end points, because the distance from their respective mid points, d_{mid}, will always be smaller than $d_{\text{interaction}}$ if the elements are equal in length. The problem is that these elements should not even be considered in contact because they are simply connected structurally. This problem posed one of the biggest challenges to integrating beam contact with the rest of the beam model. To prevent these cases of illegal contact, the solution consists of checking which elements share an interface boundary and subsequently disqualifying any contact between those elements.

Contribution of the penalty force to the weak form

After detecting contact, the model must enforce the contact constraint. One can see in Figure 3.3 that the enforcement method of choice is the penalty method. As explained in Section 2.6, a penalty approach is preferred for beam elements since it does not introduce extra degrees of freedom. When the contact state is positive, the routine calculates the penalty force at the point of contact using Equation 2.96. This penalty force must then be incorporated into the weak form shown in Equation 3.20.

The way this is achieved currently, is as such: the discrete contact force at contact point $\boldsymbol{p}_c = \boldsymbol{r}(\xi_c)$ is divided over the two nearest quadrature points, q_a and q_b, of the beam element in contact following the relations in Equation 3.48.

$$\boldsymbol{f}(\xi_c) = \boldsymbol{f}_a + \boldsymbol{f}_b \tag{3.48a}$$

$$\boldsymbol{f}_a = \frac{\|q_b - \xi_c\|}{\|q_a - q_b\|}\boldsymbol{f}(\xi_c) \tag{3.48b}$$

$$\boldsymbol{f}_b = \frac{\|q_a - \xi_c\|}{\|q_a - q_b\|}\boldsymbol{f}(\xi_c) \tag{3.48c}$$

Here, ξ_c is the coordinate of the contact point in the parametric space of the beam element. $\boldsymbol{f}(\xi_c)$ is the contact force vector at the contact point. $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$ are the contact force vectors at the two nearest quadrature points. One can observe from Equation 3.48 that $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$ are scaled linearly based on the distance of q_a and q_b from the contact point, and that their sum is equal to the total contact force. This scaling is necessary because the contact point will not always be exactly in the middle of two quadrature points.

Furthermore, the moment arms from the two nearest quadrature points to the contact point $\boldsymbol{p}_c$ are calculated. These are used to calculate the equivalent moment at the quadrature points to maintain force-moment equivalency after the distribution of the contact force over the two quadrature points. When the residual of the element is calculated, the contact force contributions of each quadrature point are added to the residual as $\tilde{\boldsymbol{f}}$ (see Equation 3.20). The contact moment contributions for each quadrature point are also calculated using Equation 3.49, where $\tilde{\boldsymbol{m}}_i$ is the moment caused by the contact force $\tilde{\boldsymbol{f}}_i$ at quadrature point i, with current position $\boldsymbol{r}_i$.

$$\tilde{\boldsymbol{m}}_i = (\boldsymbol{p}_c - \boldsymbol{r}_i) \times \tilde{\boldsymbol{f}}_i \tag{3.49}$$

Only the perpendicular components of the moment contributions are added to the weak form, because the beam formulation is torsion-free. $\tilde{\boldsymbol{m}}_{i,\perp}$ is calculated using Equation 3.10. With this, the contact contributions to $\tilde{\boldsymbol{m}}_\perp$ can be calculated, and added to the weak form in Equation 3.20. After assembling the residual of all elements, one obtains the residual force vector seen in Equation 3.26.

Alternative implementation of the penalty force contributions to the weak form

However, because the contact forces are concentrated forces, there is actually an alternative way of incorporating the penalty force contribution into the residual vector. The work contribution of a

singularity force $\boldsymbol{P}$ is given by Equation 3.50.

$$W = \int_{\Omega_h} \boldsymbol{P}\delta(\boldsymbol{p}_c(\xi_c)) \cdot \boldsymbol{\delta r} ds \tag{3.50}$$

Here, $\delta(\boldsymbol{p}_c)$ is the Dirac delta function centred at the contact point. Using the sifting property of the Dirac pulse function, $\int_{-\infty}^{\infty} f(x)\delta(a)dx = f(a)$, Equation 3.50 evaluates to Equation 3.51.

$$W = \boldsymbol{P} \cdot \boldsymbol{\delta r}(\xi_c) \tag{3.51}$$

Here, the fact that $\boldsymbol{r}$ is a function of the arc-length parameter s and thus a function of the natural coordinate ξ was used. Using Equation 3.52, and recalling that $\boldsymbol{\delta x}$ are the variations of the nodal degrees of freedom, the nodal force vector contribution of the singularity force $\boldsymbol{P}$ is given as Equation 3.53.

$$\boldsymbol{\delta r}(\xi) = \boldsymbol{N}(\xi)\boldsymbol{\delta x} \tag{3.52}$$

$$\boldsymbol{F} = \boldsymbol{N}^T(\xi_c)\boldsymbol{P} \tag{3.53}$$

In Equation 3.52 and in Equation 3.53, $\boldsymbol{N}$ represents the shape functions stacked so that the mentioned equations become vector equations. One could simplify Equation 3.53 to just $\boldsymbol{F}_a = N_a(\xi_c)\boldsymbol{P}$ for $a = 1, ...4$. Now, if one has looked at the expression of the external nodal force vector, $\boldsymbol{f}_a^{ext}$, in Appendix A of Kota et al. (2024) [16], then one may recognise that Equation 3.53 resembles the discrete force contributions on the Neumann boundary conditions, $\left[N_a \bar{\boldsymbol{f}}\right]\big|_{\partial_{N_f}\Omega_h}$.

Equation 3.53 means that the contributions of the contact forces to the residual force vector can be added to the residual force nodal field, rather than the residual force quadrature field. This would better preserve the actual discrete force since it need not be distributed over the nearest quadrature points first. As it stands now, this approach has not been implemented yet, and it makes for a great recommendation for future work. That said, adding the contact force as nodal forces does introduce an issue when one wants to also implement the line-contact or the hybrid contact formulation. The contact force for line-contact is a distributed force, which makes it much better suited for the first mentioned approach given that the distributed contact force will already be represented as a quadrature field.

Friction
The numerical experiments conducted in this thesis only consider frictionless contact, meaning the contact forces comprise solely the normal forces caused by the contact event. However, implemented in SUMMIT is a model for the tangential contact forces based on the Coulomb friction model. This model requires an extra material parameter in the form of a friction coefficient. For the sake of simplicity, the computational framework treated in this study does not take friction into account, but it should be stated that should one want to take the next step and include frictional contact forces in the resultant contact force vector, then that is certainly encouraged. It would involve determining said friction coefficient experimentally, but once this parameter has been obtained, the tangential contact forces can be calculated using the models for frictional contact developed in [97] and [98]. At that point, the tangential contact force can simply be added to the normal contact force vectorially to obtain the resultant contact force vector.

3.1.8. Tying contact with the system dynamics

Subsection 3.1.7 covers how the contact model finds contact cases and calculates the contribution of the contact force at each contact event to the residual force vector. This subsection addresses how the contact model fits into the time marching scheme used to solve the system dynamics.

The time stepping scheme has already been explained in Subsection 3.1.4, but in short, each time step consists of the following stages: (1) the nodal degrees of freedom at time t_{n+1}, $\boldsymbol{x}^{n+1}$, are predicted using Equation 3.27, (2) with x^{n+1}, the residual force vector is calculated at t_{n+1} using Equation 3.26, (3) with Equation 3.28, the accelerations at time t_{n+1}, $\ddot{\boldsymbol{x}}^{n+1}$ are calculated, (4) the velocity at time t_{n+1}, $\dot{\boldsymbol{x}}^{n+1}$ are computed.

The contact model is incorporated into the time marching scheme via an intermediate step. This intermediate step is inserted between stage (1) and stage (2) above. As such, the contact model uses the nodal degrees of freedom at time t_{n+1} to find any possible contact forces at time t_{n+1}. Then, when the residual is assembled, the contact forces are added to the residual as explained in Subsection 3.1.7.

Tying the contact model to the system dynamics presented another challenge to integrating contact with fracture behaviour. The problem came in the form of propagating the contact dynamics to the adjacent elements of a beam element in contact when interfaces are present. In the discontinuous Galerkin formulation, adjacent elements do not share nodes. As explained in Subsection 3.1.3, internal nodes are duplicated to allow discontinuities at element interfaces. As such, the interface elements play a crucial role in maintaining compatibility when there is no fracture. It is through the DG interface loads from Equation 3.20 that the interface elements ensure that contacting beam elements make the connected beam elements deform with them rather than fly off as though they were untethered. Because of this, it is vital that the weak form implementation properly communicates the weak form contributions of contact to the interfaces too. This way, the interface elements can maintain compatibility during contact.

3.1.9. Key assumptions

The list below contains the key assumptions made to offer some perspective on the unified contact and fracture framework treated by Section 3.1. Showing the assumptions before presenting the results helps with interpreting the findings. As such, the unified inter-element contact and fracture finite element framework for beam-lattice metamaterials operates under the following assumptions.

- The beam elements are considered slender and they are formulated as torsion-free geometrically nonlinear Kirchhoff beams (Subsection 3.1.1).
- A hyperelastic constitutive law is used for simplicity.
- A linear traction-separation law forms the basis of the cohesive zones (Subsection 2.7.3).
- The linear decay is governed by two material parameters: the effective fracture energy, G_c, and the cohesive strength, σ_c (Subsection 2.7.3).
- Fracture can only occur between elements.
- The time marching scheme utilises special mass lumping to save computational time (Subsection 3.1.4).
- Beam element pairs that share an interface, cannot enter contact with each other (Subsection 3.1.7).
- Contact is assumed to be frictionless (Subsection 3.1.7).

Wherever possible, a reference to the corresponding part of this thesis discussing the assumption is provided.

3.2. Modelling a truss-lattice structure for impact problems

The logical next step of the methodology is addressing how the framework synthesised in Section 3.1 can be applied to model impact problems involving lattice structures, as formulated by subtasks 2 and 3. After all, the model governing the dynamic behaviour of the lattice struts has been treated, but what must still be covered is the connection of these struts in the lattice. At first glance, this may seem like a simple case of stacking unit cells to create the lattice structure, but there are some important modelling choices involved that deserve mention.

3.2.1. The unit cell

As explained in Section 2.1, lattice structures comprise unit cells. More specifically, and more to the point for this thesis, truss-lattice structures can be built from unit cells which themselves are interconnected networks of struts and joints. Examples of truss-based unit cells have been shown in Subsection 2.1.2, however, Figure 4.1 depicts another example of a unit cell.

It should be noted that unlike real manufactured unit cells, the modelled unit cells are ideal structures without material or geometric defects. Likewise, the lattice structure is assembled from a perfectly repeating packing of unit cells. Lattice structures in real life will always have some kind of manufacturing

defect in the constituent material, the geometry of a unit cell, the packing of the unit cells or all of them. That said, if so desired, it is possible to model geometric defects by stochastically perturbing some of the nodes of a unit cell or even the whole lattice structure.

The members of each unit cell are modelled with beam elements, so the dynamics of the struts can be described by the framework, but that still leaves the joints. The joints are not represented as elements, instead, they are represented as constraints in the framework. Subsection 3.2.2 takes a deeper look at the joints.

3.2.2. Modelling joint behaviour

As eluded to earlier, the joints are represented as constraints on the kinematics of the structure. Generally, one can model the joints of a truss structure in two ways: with pin joints and with rigid joints. Pin joints allow no transmission of bending moments between joined members [99] while rigid joints do. More specifically, rigid joints fix the relative orientations of the members at the joints [100], allowing no relative rotation at all. Both types of joint constrain the members connected to a joint to remain attached to that joint.

Before the constraint equations for the joints are presented, it is instructional to define some terminology to facilitate the process of explaining the constraint equations. Following the derivation by Kota et al. (2026) $\mathcal{J}$ is defined as the set of joints in the lattice. For every joint $j \in \mathcal{J}$, one beam, identified with l_j is distinguished as the leader beam, and its deformation is not constrained. The other beams connected to the same joint j make up the set $\mathcal{F}_j$ of follower beams. These beams are constrained by the kinematics of the leader beam [100].

For pin joints, only Equation 3.54 is imposed. This leaves the rotations unconstrained.

$$\boldsymbol{\delta r}_f(s_f) = \boldsymbol{\delta r}_{l_j}(s_{l_j}) \qquad \forall j \in \mathcal{J}, \quad \forall f \in \mathcal{F}_j \tag{3.54}$$

Equation 3.54 requires that the admissible variations of the centreline positions of each follower beam coincides with the centreline position of the leader beam at each joint. $\boldsymbol{r}_{l_j}$ and $\boldsymbol{r}_f$ are the parametrisations of the leader and followers centrelines, respectively, and s_{l_j} and s_f represent the corresponding values of each beam's arc-length parameter at joint j.

Rigid joints require a second constraint equation in addition to Equation 3.54 to constrain the rotations at the joints. Because of the beam formulation used (Subsection 3.1.1), it is not possible to enforce the rotational constraints directly by requiring the tangent variations of the centreline, $\boldsymbol{\delta r}'_f(s_f)$ and $\boldsymbol{\delta r}'_{l_j}(s_{l_j})$, to be equal to each other analogous to Equation 3.54. This is because constraining the tangent variations also means constraining the axial stretch of the beams, which should be kept independent from each other at the joints. Rather than constraining the tangent variations of $\boldsymbol{r}$, the rotational variations are constrained. The reformulation of the constraint results in Equation 3.55.

$$\boldsymbol{\delta\theta}_f(s_f) = \boldsymbol{\delta\theta}_{l_j}(s_{l_j}) \qquad \forall j \in \mathcal{J}, \quad \forall f \in \mathcal{F}_j \tag{3.55}$$

When Equation 3.55 is combined with Equation 2.55 and Equation 2.34, it can also be expressed in terms of the tangents:

$$\frac{1}{||\boldsymbol{r}'_f||}\, \boldsymbol{g}_{1,f} \times \boldsymbol{\delta r}'_f = \frac{1}{||\boldsymbol{r}'_{l_j}||}\, \boldsymbol{g}_{1,l_j} \times \boldsymbol{\delta r}'_{l_j}, \qquad \forall j \in \mathcal{J}, \quad \forall f \in \mathcal{F}_j \tag{3.56}$$

In Equation 3.56, the explicit dependence of $\boldsymbol{r}'$ on the arc-length coordinate s has been dropped for conciseness.

When the cross product of Equation 3.56 is taken with $\boldsymbol{g}_{1,f}$, and using the fact that $\boldsymbol{g}_{1,f} \cdot \boldsymbol{\delta r}'_f = \delta||\boldsymbol{r}'_f||$, Equation 3.56 can be rewritten as the following equation.

$$\boldsymbol{\delta r}'_f = \delta||\boldsymbol{r}'_f||\, \boldsymbol{g}_{1,f} - \frac{||\boldsymbol{r}'_f||}{||\boldsymbol{r}'_{l_j}||}\, \boldsymbol{g}_{1,f} \times \left(\boldsymbol{g}_{1,l_j} \times \boldsymbol{\delta r}'_{l_j}\right) \qquad \forall j \in \mathcal{J}, \quad \forall f \in \mathcal{F}_j \tag{3.57}$$

Equation 3.57 defines the admissible variation of each follower beam tangent at joint j as a function of its axial stretch variation $\delta||\boldsymbol{r}'_f||$, which as mentioned earlier is not constrained, and the variation of the leader beam tangent $\boldsymbol{\delta r}'_{l_j}$. This expression is a nonlinear kinematic constraint.

The lattice structures investigated in this thesis all used rigid joints. Pin joints represent idealised constraints that do not restrict member rotations. However, real joints involve some rotational stiffness that allows for transmitting bending moments between members [101]. Therefore, rigid joints are the preferred choice of constraint. However, it should be said that real joints do not fully constrain rotations either because they would have a finite stiffness, meaning some relative rotation is possible between joined members as a result of joint deformation.

3.2.3. Resolving contact between a rigid impactor and the lattice

Related to modelling truss-lattice structures for impact problems is resolving contact with an external impactor. After all, the framework described in Section 3.1 handles beam contact, but the impactor used in the main numerical experiment of Chapter 5 is modelled as a spherical rigid body. Therefore, the impact problem also requires a suitable model for contact between a rigid impactor and the lattice.

Similar to the joints, the impactor itself is not represented as elements, but rather it is represented as a boundary condition. One can think of the rigid contact boundary condition as a slightly more complex force boundary condition which is time and position dependent. The contact force is computed according to the following routine in Figure 3.4.

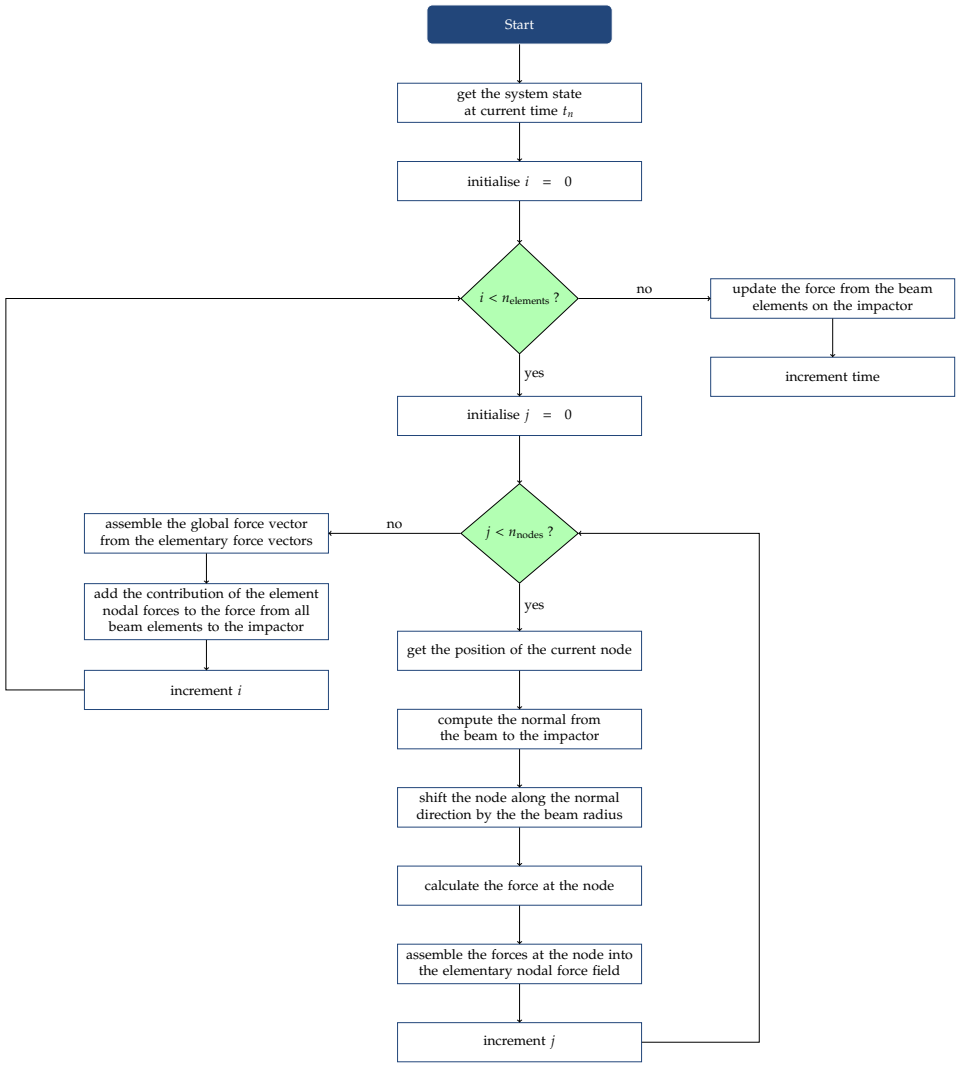


Figure 3.4: A flowchart of the rigid contact routine.

Similar to the contact force between beam elements, the contact force between a body and a rigid body is given by the equation below following Wriggers [82] in the one dimensional case.

$$F_c = -\epsilon c(u) \quad \text{for } c(u) > 0 \tag{3.58}$$

Here, F_c is the contact force, ϵ is the penalty parameter, and $c(u)$ is the interpenetration. Because the contact force is calculated by looping over each node of the lattice, one can slightly modify Equation 3.58 to represent the contact force at each node, leading to Equation 3.59.

$$F_{ci} = -\epsilon c(u)_i \quad \text{for } c(u)_i > 0 \tag{3.59}$$

F_{ci} is the contact force at node i. A 2D schematic of contact between a spherical impactor and a simplified lattice structure can be seen in Figure 3.5.

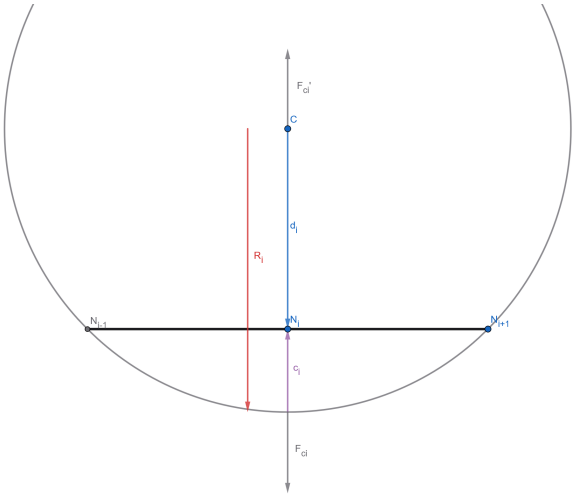


Figure 3.5: A 2D schematic of contact between the rigid impactor and a beam structure made in GeoGebra. Node N_i causes a penetration of c_i, resulting in contact force F_{ci} on node N_i and an opposite contact force F'_{ci} on the impactor.

Extending Equation 3.59 to the 3D case results in the vector equation below.

$$\boldsymbol{F_{ci}} = -\epsilon \boldsymbol{c}(\boldsymbol{u_i}) \quad \text{for } ||\boldsymbol{d}(\boldsymbol{u_i})|| < R \tag{3.60}$$

In Equation 3.60, R ($[m]$) is the radius of the spherical impactor and $\boldsymbol{d}(\boldsymbol{u_i})$ is the distance vector, defined as the difference between the centre of the rigid impactor ($\boldsymbol{u}_{\text{impactor}}$) and the i-th node of the lattice structure ($\boldsymbol{u_i}$), as given by Equation 3.61.

$$\boldsymbol{d}(\boldsymbol{u_i}) = \boldsymbol{u_i} - \boldsymbol{u}_{\text{impactor}}. \tag{3.61}$$

Next, the expression of the interpenetration $\boldsymbol{c}(\boldsymbol{u_i})$ is given by Equation 3.62
The value of the penetration $\boldsymbol{c}(\boldsymbol{u_i})$ can be found as Equation 3.62.

$$\boldsymbol{c}(\boldsymbol{u})_i = \boldsymbol{d}(\boldsymbol{u_i}) - \boldsymbol{R_i} \tag{3.62}$$

Here, $\boldsymbol{R_i}$ is the radius vector in the direction of the distance vector $\boldsymbol{d}(\boldsymbol{u_i})$. It defined as Equation 3.63. Pointing the radius into the direction of $\boldsymbol{d}(\boldsymbol{u_i})$ ensures that the interpenetration formula works for non-incident contact between a node and the impactor.

$$\boldsymbol{R_i} = R\frac{\boldsymbol{d}(\boldsymbol{u_i})}{||\boldsymbol{d}(\boldsymbol{u_i})||}. \tag{3.63}$$

With this, the contact force at each node i, $\boldsymbol{F_{ci}}$, can be computed. After looping over all elements and nodes, the global contact force vector can be assembled, and this is added to the external force nodal field $\boldsymbol{f}^{ext}$ and thereby the residual force vector $\boldsymbol{R}$ of the entire system as described in Equation 3.26. As for the force imparted onto the rigid impactor, each nodal force $\boldsymbol{F_{ci}}$ is reflected and summed over all nodes to find the total force acting on the impactor. In equation form:

$$\boldsymbol{F_c}^{\text{impactor}} = \sum_{i=1}^{N} -\boldsymbol{F_{ci}} \tag{3.64}$$

Here, N signifies the number of nodes in the system.

3.2.4. Parallelisation

Up until now, the problem has only been considered from a single CPU point of view. However, upscaling a model from a single unit cell to upwards of hundreds of unit cells means expanding the model from at most a couple of hundred nodes to a model encompassing tens of thousands of nodes. Models of this scale necessitate parallelisation to perform meaningful simulations in a reasonable time span. When the problem is parallelised, it is partitioned and handed to multiple processors for

simultaneous computations. This affects the simulation on two fronts: the partition of the mesh and the parallelisation of the solver code.

Partitioning the mesh
The mesh is partitioned by dividing it into subdomains, as can be seen in Figure 3.6. Partitioning of the mesh is taken care of by METIS [102], which is a program that employs graph partitioning algorithms like multilevel recursive-bisection, multilevel k-way and multi-constraint partitioning schemes.

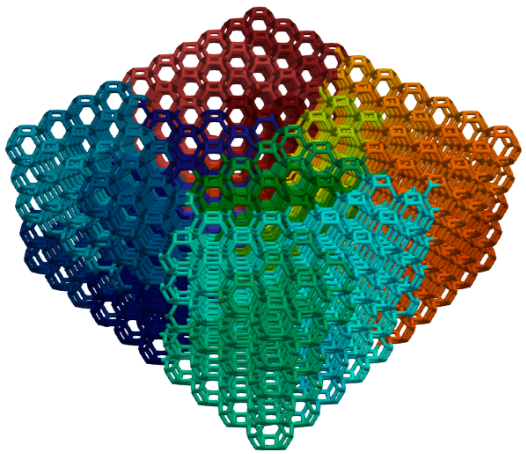


Figure 3.6: The partitions of the lattice for parallel computing.

Parallelising the code
Most of the solver code, built in SUMMIT, has been implemented with parallel computing in mind. This enables the solver engine to divide the calculations over multiple processing units. In the case of SUMMIT, the Message Passing Interface (MPI) is utilised to facilitate this. The parallelisation scheme of the solver routine has each computing node carrying out the solver routine outlined in Subsection 3.1.4 on their assigned mesh partition, for every time step. The results at each time step are communicated between partitions to ensure that the solution remains consistent between partitions. This way, the results over the entire problem stay synchronised for all partitions.

Contact between the rigid impactor and the lattice is parallelised by giving each partition the same rigid impactor, essentially running the rigid contact routine described in Subsection 3.2.3 for each partition at the same time. The contact force on the impactor in each partition is also communicated over each partition so that the total force imparted onto the impactor is actually equal to the total force the whole structure would apply on the impactor. With the contact force synchronised, the impactor kinematics are also automatically synchronised.

One may have noticed the opening line that "most of the solver code" has been implemented with parallel computing in mind. Indeed, one major part of the solver code has not been implemented with parallelisation in mind, which is the beam contact model. As it stands now, the contact step of the solver can only resolve contact within a partition, but not between partitions. This has to do with the fact that the contact routine needs to know of all elements to find element pairs, but when the mesh has been partitioned, the system on one partition only knows of the elements on that one partition.

One work around could be to pass along the complete set of elements between all partitions, but as one can imagine, this will severely reduce the efficacy of the parallelisation because each partition will now look for potential element pairs between each partition. A better solution more compatible with

parallel computing would be to redesign the way the potential contact pairs are determined which does not involve searching over every element set.

For the purposes of this study, the fact that inter-partition contact is not resolved is conceded, but one can make the argument that for capturing densification behaviour in impact, the stiffening effect of the struts of the compressed cells contacting each other is primarily intra-partition contact as long as the partitions are large enough enough potential contact events. Inter-partition contact events gain more emphasis when impact causes tons of fragments to scatter at high velocities, which can further damage the structure beyond initial impact with the impactor. Naturally, true parallelisation of the contact model is a strong recommendation for future work.

3.2.5. Key assumptions

In a similar vein to Subsection 3.1.9, a list with the key assumptions going into modelling a truss-lattice structure for impact problems is presented below. These assumptions mainly involve modelling choices of the impact problem and the lattice structure, and are therefore mostly separate from the computational framework described in Section 3.1.

- The lattice structure is composed of a perfectly repeating pattern of unit cells (Subsection 3.2.1).
- If desired, geometric perturbations can be applied to the lattice nodes to mimic geometric defects (Subsection 3.2.1).
- Joint behaviour is modelled as a kinematic constraint on the joined members (Subsection 3.2.2).
- The lattice joints are modelled as rigid joints (Subsection 3.2.2).
- The impactor in the impact problem is assumed to be rigid and spherical (Subsection 3.2.3).
- For parallel computing, the inter-partition contact events are conceded (Subsection 3.2.4).

3.3. Quantifying energy components

Section 3.1 and Section 3.2 treated the modelling side of this study, however, in order to test the functionality of the computational framework as outlined by subtask 2, the energy balance of the system must be tracked at every time. Doing so would provide insights into the system response during and after impact. That said, tracking the energy components for the lattice can also be useful for the application described in subtask 3 since they can be used to verify energy balance of the system and to relate the impact response to the different energy contributions of the lattice. As such, this section covers which energy components are tracked, and how they are quantified.

3.3.1. Energy balance of a mechanical system

The balance of energy of a mechanical system is a fundamental concept in the analysis of mechanical systems, their deformation and their motion. Often times, the time history of the energy balance and its components speaks volumes about the response of a mechanical system under a load.
In this study, the energy balance is formulated using the total energy E_{tot}, as seen in Equation 3.65, and from there, Equation 3.66.

$$E_{tot} = E_{\text{kin}} + W_{\text{int}} - W_{\text{ext}} \tag{3.65}$$

$$E_{mech} = E_{\text{kin}} + W_{\text{int}} \tag{3.66}$$

Here, E_{kin} is the kinetic energy at a given time. Equation 3.65 is meant to represent the cumulative total energy, kinetic energy, internal work and external work. The sign of W_{ext} was chosen as negative since this term represents external work. E_{mech} in Equation 3.66 is defined as the total mechanical energy of the system. Conservation of mechanical energy is then satisfied if E_{mech} is constant. Similarly, balance of energy is satisfied when E_{tot} is constant. Equation 3.65 and Equation 3.66 can be plotted against time, and can be immediately interpreted since they represent cumulative quantities.

Strictly speaking, for the purposes of mechanical energy balance, Equation 3.65 is sufficient to determine at a glance whether a system conserves energy. When applied to the impact problem, one would expect that the total energy decreases as a result of the dissipation mechanisms touched upon in Subsection 2.3.4. However, this makes it impossible to distinguish what kind of dissipation mechanism was at play because all their effects are now confounded into the energy deficit. If one would like to

quantify the energy dissipation of specific dissipation modes, then tracking their associated energy is a must.

To this end, Equation 3.65 and Equation 3.66 can be modified by adding extra terms representing the energy contributions from the different dissipation modes. Since the computational framework only considers fracture and contact, and not plasticity or thermal effects, Equation 3.65 and Equation 3.66 are only modified by adding an interface energy term to represent the work needed to cause fracture in the cohesive zones.

$$E_{tot} = E_{\text{kin}} + W_{\text{int}} + W_{\text{frac}} - W_{\text{ext}} \tag{3.67}$$

$$E_{mech} = E_{\text{kin}} + W_{\text{int}} + W_{\text{frac}} \tag{3.68}$$

In Equation 3.67, W_{frac} is the interface energy. Its sign is chosen as positive since absence of damage means the fracture energy is 0, and the total energy is equal to the total energy from Equation 3.65. If there is damage, then $W_{\text{frac}} > 0$, but the total energy E_{tot} still remains constant (assuming the only dissipation mechanism is crack growth). This way, if there is actually another untracked dissipation mechanism at play, then one can immediately tell from E_{tot}. In a sense, adding energy components to Equation 3.67 is a means of tracking said energy components. The next subsections will cover each energy component of Equation 3.67.

Before moving on to the individual energy components, Equation 3.67 should be slightly rewritten to increase clarity for the case when an impactor is involved.

$$E_{tot_{\text{system}}} = E_{tot_{\text{FEM}}} + E_{tot_{\text{impactor}}} = E_{\text{kin}_{\text{FEM}}} + E_{\text{kin}_{\text{impactor}}} + W_{\text{int}} + W_{\text{frac}} - W_{\text{ext}_{\text{FEM}}} - W_{\text{ext}_{\text{impactor}}} \tag{3.69}$$

Equation 3.69 describes the total tracked energy of the system, meaning both the lattice structure and the impactor. No effort was made to distinguish between the internal work or interface energy of the finite lattice structure and the impactor, because the impactor is a rigid body. This naturally leads to the expressions for the total energies of the finite element system and the impactor in isolation, Equation 3.70 and Equation 3.71.

$$E_{tot_{\text{FEM}}} = E_{\text{kin}_{\text{FEM}}} + W_{\text{int}} + W_{\text{frac}} - W_{\text{ext}_{\text{FEM}}} \tag{3.70}$$

$$E_{tot_{\text{impactor}}} = E_{\text{kin}_{\text{impactor}}} - W_{\text{ext}_{\text{impactor}}} \tag{3.71}$$

Assuming that the system consists only of an impactor and a lattice, then Equation 3.69 can be simplified to Equation 3.72 since the external work done by the impactor and the external work done by the lattice are of equal magnitudes and of opposite signs in this case. The impactor does work on the lattice during collision, so the same value of $W_{\text{ext}_{\text{impactor}}}$ also appears as $W_{\text{ext}_{\text{FEM}}}$ but with an opposite sign.

$$E_{tot_{\text{system}}} = E_{tot_{\text{FEM}}} + E_{tot_{\text{impactor}}} = E_{\text{kin}_{\text{FEM}}} + E_{\text{kin}_{\text{impactor}}} + W_{\text{int}} + W_{\text{frac}} \tag{3.72}$$

3.3.2. Kinetic energy

The kinetic energy of the finite element system is calculated using Equation 3.73.

$$E_{\text{kin}_{\text{FEM}}}(t_n) = \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2(t_n) \tag{3.73}$$

Here, N is the total number of unconstrained degrees of freedom, m_i is the lumped mass contribution of degree of freedom i, $v_i(t_n)$ is the time derivative of degree of freedom i at time t_n. Although the mass field is the lumped mass field, using enough elements means that the mass field closely approximates the real mass field. More noticeably is the lack of any rotational contributions. This has to do with the fact that torsion-free geometrically nonlinear Kirchhoff beam elements are used, which means the rotational inertia can be neglected Subsection 3.1.2. Comparing Equation 3.73 with the kinetic energy expression from Equation 3.2, one can see the similarities. Equation 3.73 is the preferred way of calculating the kinetic energy because it involves no integration over each element since the nodal velocity field is accessible to the entire system, and the lumped mass has been assembled at the very start.

That said, if one were to compute the kinetic energy by integrating the kinetic energy per unit length of a torsion-free geometrically nonlinear Kirchhoff beam from Subsection 3.1.2 over the length of each element and then summing over each element, then one would arrive at Equation 3.74.

$$E_{\text{kin}_{\text{FEM}}}(t_n) = \sum_{i=1}^{E} \int_0^{l_i} \frac{1}{2} \rho_i A_i \dot{\boldsymbol{r}}_i^T(s, t_n) \dot{\boldsymbol{r}}_i(s, t_n) ds \tag{3.74}$$

In the equation above, E is the total number of elements. $\rho_i A_i$ is the mass per unit length of element i. $\dot{\boldsymbol{r}}_i(s, t_n)$ is the velocity field of element i at arc-length position s and time t_n. Although implemented in SUMMIT, this method of calculating the kinetic energy is not used in this thesis because of the perceived advantages of using Equation 3.73 owing to it not requiring any integration and thus summation over the elements. Still, it may be a worthwhile exercise for future work to compare the kinetic energy computed using Equation 3.73 to the kinetic energy following Equation 3.74.

Now, when the rigid impactor is involved, the total system kinetic energy must also include the kinetic energy of the impactor since the impactor is not represented in the finite element system as elements. The kinetic energy of the rigid impactor is given by Equation 3.75.

$$E_{\text{kin}_{\text{impactor}}}(t_n) = \sum_{i=1}^{3} \frac{1}{2} m_{\text{rigid}} v_i^2(t_n) \tag{3.75}$$

This value can then simply be added to Equation 3.73 during post-processing to recover the full system kinetic energy.

3.3.3. Internal energy

The internal energy of the finite element system is given by Equation 3.76.

$$W_{\text{int}}(t_n) = \sum_{i=1}^{E} \Pi_{\text{int}}^i(t_n) \tag{3.76}$$

Here, E is the total number of elements and $\Pi_{\text{int}}^i(t_n)$ is the internal energy of element i at time t_n. The expression for Π_{int} is given in Equation 3.2. Equation 3.76 must be integrated over each element.

Because the impactor is a rigid body, it does not contribute to W_{int}.

3.3.4. External work

To calculate the cumulative external work of the finite element system, it is actually more convenient to calculate the increment in external work $\Delta W_{\text{int}_{\text{FEM}}}$ at time t_n. This is done as follows.

$$\Delta W_{\text{ext}_{\text{FEM}}}(t_n) = \sum_{i=1}^{N} \frac{1}{2} (\bar{f}_i(t_n) + \bar{f}_i(t_{n-1})) \Delta x_i(t_n) + \sum_{j=1}^{M} \frac{1}{2} (R_j(t_n) + R_j(t_{n-1})) \Delta x_j(t_n) \tag{3.77}$$

In Equation 3.77, N signifies the total number of Neumann boundary conditions, while M is the total number of Dirichlet boundary conditions. For the Neumann, or force, boundary conditions, an external force $\bar{f}$ is applied to degree of freedom i. The increment work contribution is found by taking the average external force and multiplying it by the change in degree of freedom i, $\Delta x_i(t_n) = x_i(t_n) - x_i(t_{n-1})$. For the Dirichlet, or displacement, boundary conditions, degree of freedom j is prescribed. The increment work contribution is computed by taking the average of the residual at degree of freedom j and multiplying it by the change in degree of freedom j, $\Delta x_j(t_n) = x_j(t_n) - x_j(t_{n-1})$. The cumulative external work can then be computed by summing each increment.

The external work on the impactor can be calculated in a similar way.

$$\Delta W_{\text{ext}_{\text{impactor}}}(t_n) = \sum_{i=1}^{3} \frac{1}{2} (F_{c_i}(t_n) + F_{c_i}(t_{n-1})) \Delta x_i(t_n) \tag{3.78}$$

In Equation 3.78, F_{c_i} is the component in the i-direction of the total contact force acting on the impactor. The average contact force in the i-direction is then multiplied by the change in position of the impactor's centre $\Delta x_i(t_n) = x_i(t_n) - x_i(t_{n-1})$. The cumulative external work is calculated by summing each increment. That said, as explained earlier and as illustrated by Equation 3.72, for the lattice-impactor problem, it is not necessary to compute the external work of the impactor and lattice since those two would cancel each other when summed.

Contribution of contact to external work

From a numerical point of view, beam contact forces can contribute to external work, if only for a very short time. When the contacting members stay in contact, this either means they have come to a halt, or they are moving in the same direction (e.g. inelastic collision or forced displacement). In both cases, the work done by the contact forces on both members is net 0 since the contact forces are opposite in direction but the displacement is 0 (in case of the former) or the same (in case of the latter). However, this ignores the effect of interpenetration. While two beam members penetrate into each other, they are subjected to contact forces, and depending on the contact penalty stiffness and the time step, it may take some time before the contact force manages to reduce the interpenetration to 0 again. Figure 3.7 shows the interpenetration g schematically in time.

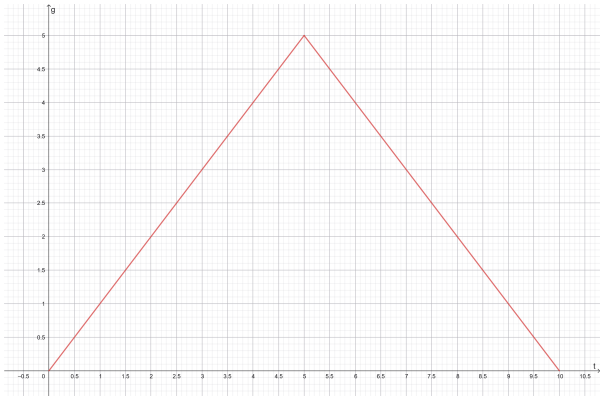


Figure 3.7: A schematic of the interpenetration of two contacting beams in time, made in GeoGebra.

Note that in reality, $g(t)$ is not a triangle wave, since this would imply the velocity is constant. However, because the contact force is proportional to g, the acceleration will not be 0 meaning the velocity cannot be constant. Still, this schematic tells one important thing: when the penalty contact approach is used, then contacting elements do not immediately return to the no-penetration state, but rather it takes time until the penetration goes back to 0. However, while this happens, the contact forces actually do work. Consider an example of two contacting beams based on the schematic Figure 3.7: just after contact ($t = 0$) and before g reaches its peak value, both beams have some displacement from their original position and are subjected to contact forces. This has been schematically described by Figure 3.8.

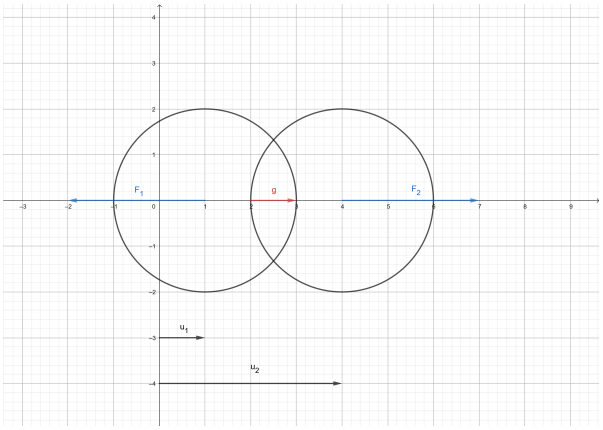


Figure 3.8: A schematic of two contacting beams, made in GeoGebra.

In Figure 3.8, the two contacting beams are represented by the two circles, effectively showing the side view. $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are the current positions of the two beams. g is the interpenetration at the current time, and $\boldsymbol{F_1}$ and $\boldsymbol{F_2}$ are the associated contact forces on the two beams. One can see that the contact forces will do work on the two beam elements. If $d\boldsymbol{u_1} = \boldsymbol{u_1}(t) - \boldsymbol{u_1}(0)$ and $d\boldsymbol{u_2} = \boldsymbol{u_2}(t) - \boldsymbol{u_2}(0)$ (here $t = 0$ means at the start of contact) are equal in magnitude and point in the same direction, then the total work of the contact forces $W_{\text{contact}} = \boldsymbol{F}_1 \cdot d\boldsymbol{u}_1 + \boldsymbol{F}_2 \cdot d\boldsymbol{u}_2$ will be net 0 because $\boldsymbol{F_1} = -\boldsymbol{F_2}$ (assuming no friction). However, $d\boldsymbol{u}_1 = d\boldsymbol{u}_2$ is not necessarily true in general for contacting beams. If one beam has constrained positions, then $d\boldsymbol{u} = \boldsymbol{0}$ for that beam. This means that $W_{\text{contact}} \neq 0$. Similarly, if one untethered beam element such as a fragment of a broken strut contacts a beam element which is still connected to adjacent elements, then the displacement of the connected beam element will be smaller than the displacement of the untethered beam element: $\|d\boldsymbol{u}_2\| \neq \|d\boldsymbol{u}_1\|$, because the connected beam element is kinematically constrained by the adjacent elements. In this case W_{contact} is also non-zero.

Now, after the peak of g the interpenetration eventually decreases back to 0 again and the beams lose contact (the right side of the triangle in Figure 3.7), the $d\boldsymbol{u}_1$ will generally point in the opposite direction to $d\boldsymbol{u}_2$ unless they move in the same direction after collision. If $d\boldsymbol{u}_1$ points in the opposite direction of $d\boldsymbol{u}_2$ while $g \neq 0$ meaning the contact forces are still non-zero, then $W_{\text{contact}} \neq 0$. This is true for when $\|d\boldsymbol{u}_2\| \neq \|d\boldsymbol{u}_1\|$ and when $\|d\boldsymbol{u}_2\| = \|d\boldsymbol{u}_1\|$. As such, the above example illustrates how the work contribution of the contact forces both before and after the maximum interpenetration can be non-zero.

This contribution of the contact forces between the beam elements of the lattice to the total energy is not tracked. However, this work contribution is mostly dependent on the time step. If the time step is sufficiently small, then the work increment caused by the contact forces becomes negligible compared to the other energy components.

3.3.5. Interface energy

The interface energy of the cohesive zones is calculated using the following equation.

$$W_{\text{frac}}(t_n) = \sum_{i=1}^{C} \tilde{W}^i_{\text{frac}}(t_n) \tag{3.79}$$

Here, C is the total number of interfaces, and $\tilde{W}^i_{\text{frac}}(t_n)$ is the interface energy of interface i at time t_n. $\tilde{W}_{\text{frac}}$ can be computed using Equation 3.80.

$$\tilde{W}_{\text{frac}}(t_n) = \begin{cases} \frac{1}{2}(f_c + f_{coh}(t_n))\Delta(t_n), & \text{if } \Delta \geq \Delta_{max} \\ \frac{1}{2} f_c \Delta_c, & \text{if } \Delta \geq \Delta_c \\ \tilde{W}_{\text{frac}}(t_{n-1}), & \text{if } \Delta < \Delta_{max} \end{cases} \tag{3.80}$$

Here, f_c is the critical cohesive force, $f_{coh}(t_n)$ is the effective cohesive force at the interface at time t_n, $\Delta(t_n)$ is the effective separation at the interface at time t_n, Δ_{max} is the maximum effective separa-

tion in the entire loading history, and Δ_c is the effective separation at which complete decohesion occurs.

Equation 3.80 essentially states that the interface energy is equal to the area under the cohesive law Equation 2.125 (see also Figure 2.44). If $\Delta_{max} < \Delta < \Delta_c$, then the area is from the origin up until Δ, which is a trapezium, hence why the formula for the fracture energy in this regime is equal to the area of a trapezium. If $\Delta \geq \Delta_c$, then complete fracture has occurred, so the interface energy is equal to the total area under the cohesive law, which is a triangle. This also implies that post fracture, the interface energy must stay constant because there is nothing left to put energy into. If $\Delta < \Delta_{max}$, the crack does not grow, meaning no energy is dissipated because of it, and as such, $\tilde{W}_{\text{frac}}$ does not change. The way Equation 3.80 is formulated captures permanent damage. Even if the interface is unloaded so that $\Delta = 0$, $\tilde{W}_{\text{frac}}$ maintains the last reached value.

In principle, the DG interface forces and moments from Equation 3.20 can contribute to the interface energy, however, in practice, the pre-fracture effective separation Δ is so small, that these loads have a negligible contribution to the interface energy.

3.3.6. Assumptions and potential sources of error

Just as with Section 3.1 and Section 3.2, the assumptions will be listed. Doing so before presenting the results may help explain potential sources of error later.

- For the lattice structure, only the kinetic energy, internal energy, external work and the interface energy are tracked (Subsection 3.3.1).
- For the impactor, only the kinetic energy and external work are tracked (Subsection 3.3.1).
- The energy contributions of plastic deformations, shock waves and heating are not tracked because these phenomena are not modelled (Subsection 3.3.1).
- The impactor is modelled as a rigid sphere so it does not contribute to the internal energy of the system (Subsection 3.3.3).
- The energy contribution of the beam contact forces is not tracked (Subsection 3.3.4).
- The interface energy is based on a linear cohesive law (Subsection 3.3.5).
- The interface energy only comprises the contribution from the cohesive forces and moments (Subsection 3.3.5).

3.4. The SUMMIT framework

As mentioned during the introduction of this chapter, the SUMMIT framework serves as the basis for the simulations and the computational framework covered in Section 3.1. It is therefore fitting to cover SUMMIT, and to this end, this section will introduce SUMMIT, highlight some of its key features, and provide a high-level overview of the SUMMIT framework is used in the simulations conducted during this study.

SUMMIT is a finite element code base, and the formulations and models described earlier have been implemented in code in SUMMIT, which is the research code of the Advanced Computing for Engineering and Science group at TU Delft led by Dr Bianca Giovanardi. Two studies referenced before, on computationally modelling fracture in geometrically exact beams [16], and on large-scale numerical modelling of fracture in truss architected materials [100], are examples of SUMMIT being applied to model computational mechanics problems. Other applications of SUMMIT include a study on large-scale simulation of fluid-driven fracture propagation [103], and simulating the dynamics of large assemblies of highly-flexible fibres immersed in viscous flow [104].

One of SUMMIT's defining features is the fact it is written in C++. This means it benefits from all the advantages attributed to C++, such as high performance, Object-Oriented Programming (OOP) support, and platform independence. The high performance of C++ is especially useful for SUMMIT since the types of applications built in SUMMIT will be of the computationally demanding kind. After all, the calling of a FEM framework is modelling complex mechanics problems. The ability to use OOP in SUMMIT lends itself to further benefits such as code reusability, improved maintainability and modularity. This becomes especially useful when one considers that SUMMIT is developed by a team, which further

emphasises the need for modularity and maintainability. It is this modularity which also gives SUMMIT the capacity to model widely different problems, from fluid-structure problems to crack-propagation problems. Lastly, the platform independence of C++ makes porting SUMMIT applications to different platforms easier. This further facilitates SUMMIT's collaborative development, and promotes the deployment of SUMMIT to high-performance computing (HPC) clusters. The latter is another hugely beneficial trait of SUMMIT since large-scale problems often necessitate the use of distributed computing.

Figure 3.9 gives an abstract overview of how the SUMMIT framework is used for the simulations, that is, in SUMMIT applications. The depicted diagram both serves to emphasise the modular nature of SUMMIT, as well as highlight some of the important modules in SUMMIT which were modified.

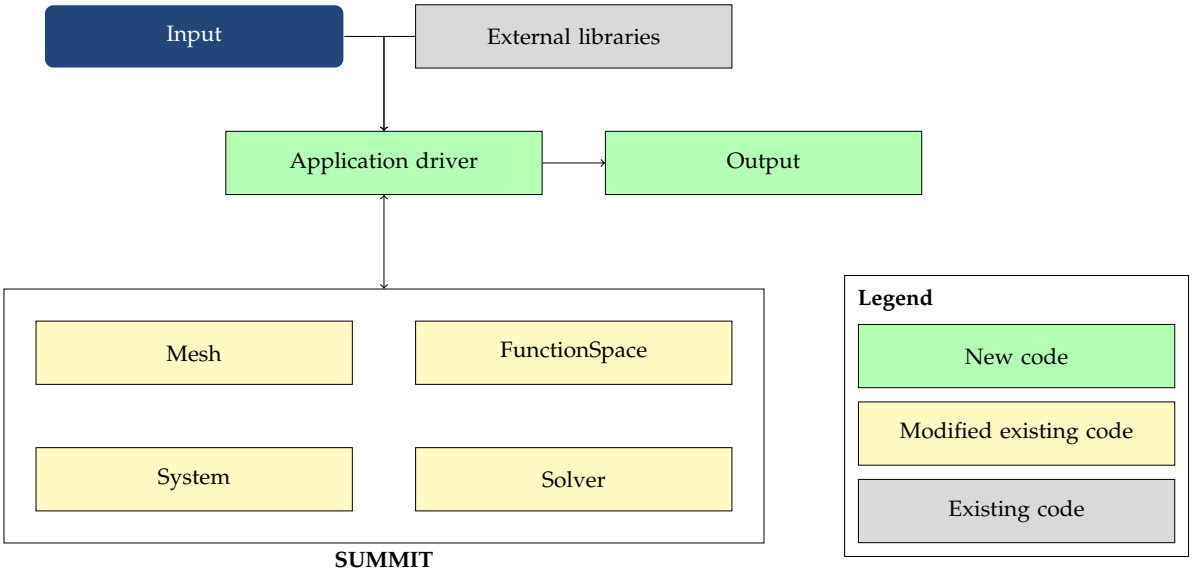


Figure 3.9: A diagram of how a SUMMIT application works.

As can be seen in Figure 3.9, SUMMIT applications are built around a driver file and the SUMMIT library itself. The meaning of the blocks in Figure 3.9 is explained below.

- **Input** is an abstraction for all the user-defined data required to make the application run. This includes material properties, system and/or solver parameters, geometric inputs or even the entire mesh file.
- **External libraries** represent the non-SUMMIT libraries used by an application. Most notable for the applications developed for this thesis is MITO. One of utilities provided by the external libraries is mesh generation.
- **The application driver** interfaces with SUMMIT and controls the simulation. Using modules from SUMMIT, the driver interacts with the inputs by discretising the mesh, creating a system, initialising a solver, and setting boundary conditions and initial conditions. After this, it marches through time to solve the system while writing the results to output files.
- **Output** is an abstraction for all the un-processed output data written by the driver. This can be in the form of VTK files, which can be visualised directly in software like Paraview, or it could be raw data written to dat or csv files.
- **Mesh** is a SUMMIT object that takes a mesh file, and splits the geometry into cells bound by vertices.
- **FunctionSpace** adds physical meaning to the geometry by taking the Mesh object and splitting it into element sets.
- **System** is the object which drives the analysis. It takes the FunctionSpace object and the material properties, and it holds the governing equations and the primary variables. The governing equation defining a System is the weak form, which has contributions from every part of the Mesh. These parts are known as Regions.

- **Solver** is responsible for solving the System. This object takes the System, and solves the weak form for the next time step based on the type of solver used.

It must be said that while some modifications were necessary to the modules mentioned above to make a beam-contact-fracture implementation work, most of the code could be repurposed so that the basic architecture of SUMMIT remains unchanged.

4

Numerical experiments

Subtask 2

Test the functionality of the unified beam-contact-fracture framework in simple numerical experiments.

Before deploying the framework to the large-scale numerical experiment of impact in a lattice structure, three test applications are presented wherein the computational model is applied to simpler test cases than the final experiment. The purpose is to demonstrate the performance of the unified inter-beam contact and fracture model in stages. In doing so, each stage can isolate the performance of the computational framework in a certain area, facilitating debugging and further software development. After treating all test applications and their results, this chapter ends on some comments on said test results.

The first test case considers a unit inside body-centred cubic (insideBCC) cell compressed to test fracture and inter-beam contact under quasi-static loading. Case 2 features a unit insideBCC cell subjected to an initial velocity front to investigate fracture and energy balance under dynamic loading. The third case models the collision of two unit insideBCC cells with the intention of testing contact and energy balance during collision.

4.1. Case I: compression of a unit insideBCC

The first simulation case models a unit insideBCC cell uniformed compressed under constant displacement rate. The purpose of this test case to test the fracture and contact behaviour of the cell struts when the cell is gradually compressed. Before jumping into the results, some time will be taken to explain the set-up and the insideBCC unit cell.

4.1.1. Set-up and the insideBCC unit cell

The unit cells used in these test cases are all of the type insideBCC. The unit cell geometry can be seen in Figure 4.1. It resembles a BCC cell superimposed with a simple cubic cell.

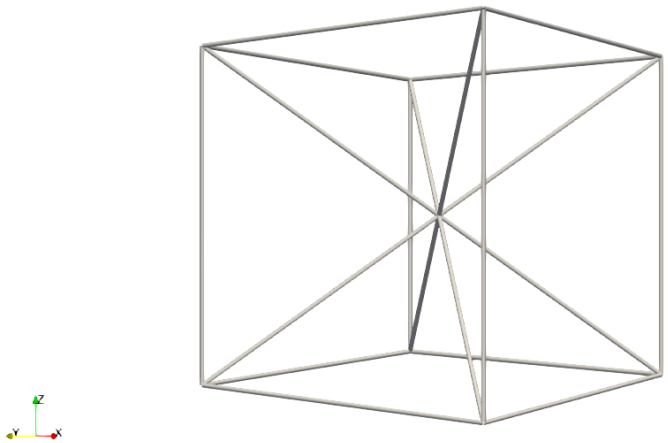


Figure 4.1: The unit insideBCC cell used throughout the test applications. Also the set-up of case I.

The unit insideBCC cell used throughout the test cases has dimensions 0.1 [m] by 0.1 [m] by 0.1 [m]. The simulation parameters have been summarised in Table 4.1.

Table 4.1: Simulation parameters for case I: the compression of a single insideBCC cell.

Parameter	Symbol [unit]	Value
Reference strut length	L_0 [m]	0.1
Young's modulus	E [MPa]	0.5
Strut radius	r [m]	0.001
Strut mass density	ρ [kg/m^3]	1000
DG interface penalty	β_p and β_t [-]	1000
Cohesive strength	σ_c [Pa]	9000
Fracture energy	G_c [N/m]	1
Mode-mixity parameter	α [-]	1
Displacement factor	R_{disp} [-]	-0.4
Displacement rate	r_{disp} [m/s]	0.005
Contact penalty stiffness factor	R_{contact} [-]	10.0
Nº elements per strut	N [-]	10
Time step	Δt [s]	1.0e-5

In Table 4.1, β_p and β_t are the DG penalty parameters used in Equation 3.20. σ_c and G_c are used in the cohesive law of the cohesive zone elements Equation 2.125 and Equation 2.126. α is used to calculate the effective separation and the equivalent interface force Equation 2.123 and Equation 2.128. The displacement factor R_{disp} is the fraction of the initial height by which the cube is displaced, see Equation 4.1.

$$R_{\text{disp}} = \frac{D}{L_0} \tag{4.1}$$

Here, D is the final displacement of the cube in the z-direction. This simulation is executed at constant displacement rate and fixed final displacement. Since the time step is also fixed, this means the final time is determined by the final displacement value, and the number of time steps is determined from the displacement rate, time step and final displacement. This choice of simulation driver is preferred over a fixed number of time steps and free displacement rate since it allows for exact control of the displacement rate. The number of time steps is calculated using Equation 4.2.

$$N_{\text{time steps}} = \text{floor}\left(\frac{|D|}{r_{\text{disp}}\Delta t}\right) \tag{4.2}$$

Here, floor() represents the floor function in C++, which returns the largest integer value less than or equal to the argument. With the number of time steps known, the forced displacement at each time step, n, is given by Equation 4.3.

$$d = L_0 + \frac{n}{N_{\text{time steps}}}D \tag{4.3}$$

These displacements are applied to the top nodes of the cube as Dirichlet boundary conditions. Essentially, this means all the top nodes of the cube have a prescribed displacement in the z-direction that is incremented each time step until the final height is reached. The bottom nodes of the unit cell are fixed. Note that the simulations are not run with gravity. Lastly, the contact penalty stiffness factor R_{contact} is used as a scaling factor to calculate the penalty contact stiffness using Equation 4.4.

$$k_p = R_{\text{contact}}k_{eq} \tag{4.4}$$

Here, k_{eq} is a reference stiffness. In this case, the equivalent axial spring stiffness of the vertical struts is used, given by Equation 4.5.

$$k_{eq} = \frac{EA}{L} \tag{4.5}$$

Using a scaling factor for the contact penalty rather than the actual contact penalty stiffness itself makes tuning the penalty stiffness more convenient. With these equations, the value of the penalty stiffness, k_p, is calculated to be 157.08 N/m.

Figure 4.2 depicts a schematic of case I, showing how the cube is constrained both at the bottom and at the top.

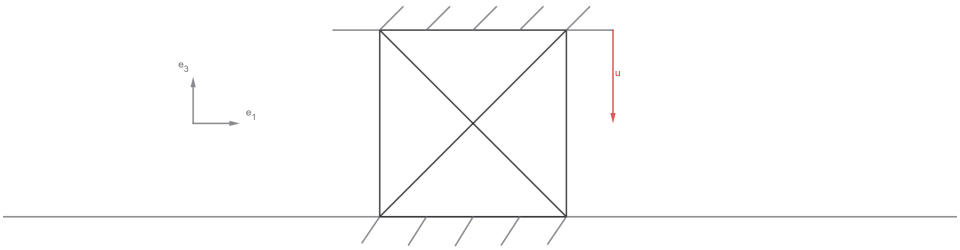
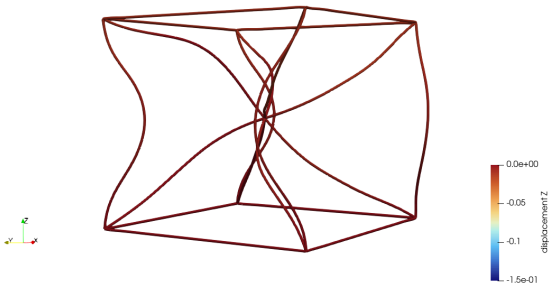


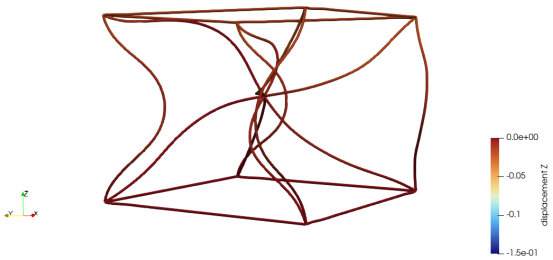
Figure 4.2: A schematic of the loading of test case I, made in GeoGebra.

4.1.2. Results

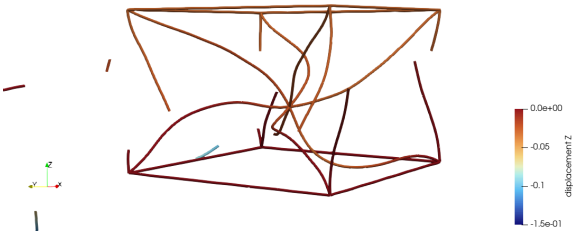
Four frames have been extracted from the simulation run and can be seen below in Figure 4.3.



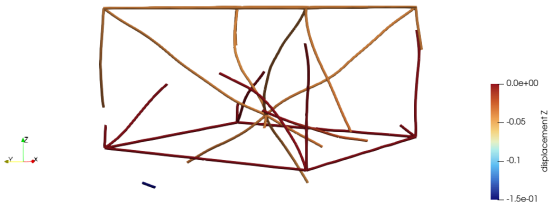
(a) Simulation frame for case I captured at approximately 2.0 s.



(b) Simulation frame for case I captured at approximately 4.0 s.



(c) Simulation frame for case I captured at approximately 6.0 s.



(d) Simulation frame for case I captured at approximately 8.0 s.

Figure 4.3: Four simulation frames for case I. Each frame corresponds to approximately 0.04 s.

Figure 4.3d is the last frame of the simulation, so it also represents the final state of the system. There are two things of note: fracture occurs first at the vertical struts, and when all struts connecting the top members to the bottom members have fractured, the top half of the cube is freely displaced without being resisted by the bottom half the cube (aside from the occasional contact events between the struts). The fact that fracture occurs at the vertical struts first is as expected because these members are subjected to the largest bending moments owing to their large curvature as seen in Figure 4.3a. The second observation is a consequence of the relatively low level of interconnection in an insideBCC cell. Aside from the vertical struts and the joint in the centre, there are not many ways the top half of the cube is connected to the bottom half, so structural integrity is lost quite quickly.

The energy components and the total mechanical energy of the system have been plotted as Figure 4.4.

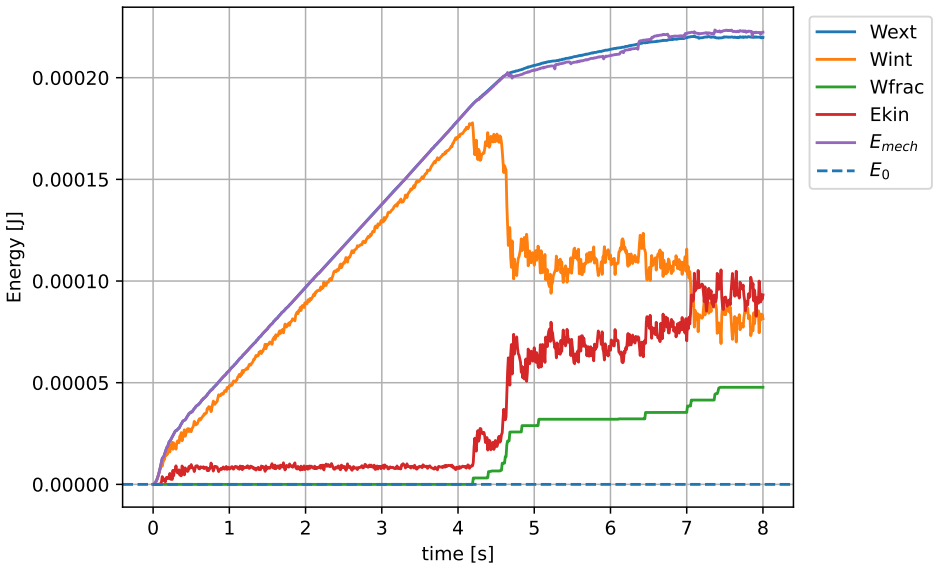


Figure 4.4: The energy balance of the system from case I for $R_{\text{disp}} = -0.4$ and $r_{\text{disp}} = 0.005$.

Figure 4.4 shows the external work, W_{ext}, the internal energy, W_{int}, the interface energy, W_{frac}, the kinetic energy, E_{kin}, and the total system mechanical energy, E_{mech}, as functions of time. It also shows the initial energy at time $t = 0$ s, but for a displacement load simulation, this is 0.0 J. One can see that E_{mech} is generally equal to W_{ext}, which implies that E_{tot} is roughly 0.0 J and constant, thereby satisfying energy balance. Keep in mind that in this forced displacement case, energy is actively being injected into the system due to the prescribed displacement, so E_{mech} cannot possibly be constant.

After $t = 4$ s, the kinetic energy of the system starts rising because of the fragmented elements that broke off from the fractured struts. After $t = 7$ s the struts connecting the centre joint to the bottom half of the cube break, which also causes a jump in the kinetic energy as the strain energy in the diagonal struts is released. The external work plateaus after this point because the forced displacement does not actually do work on the structure any more since the load paths through the bottom diagonal struts have been interrupted. As one can see from Figure 4.4, the total mechanical energy follows suit since no more new energy is injected into the system.

Figure 4.5 is the energy plot when the same experiment is run but with a displacement rate of 0.05 m/s.

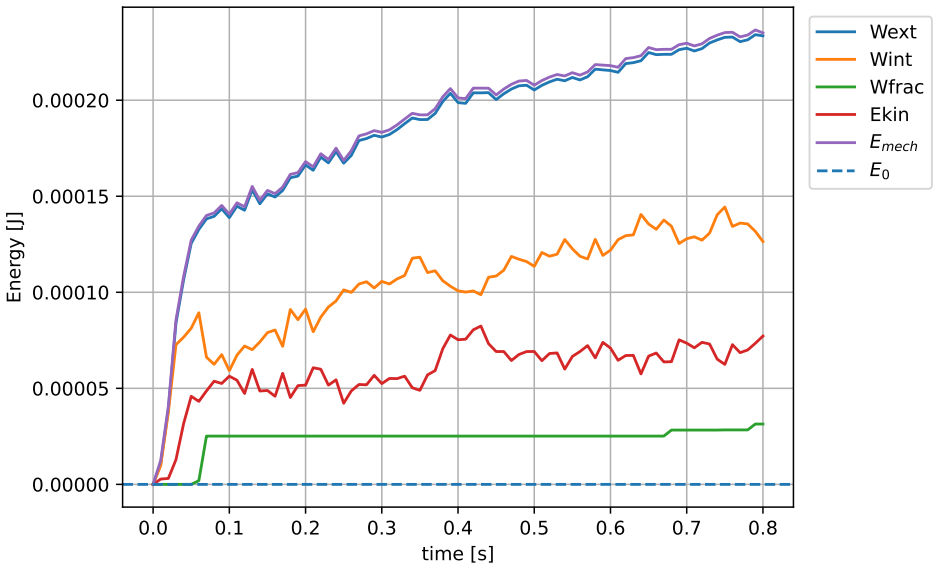


Figure 4.5: The energy balance of the system from case I for $R_{\text{disp}} = -0.4$ and $r_{\text{disp}} = 0.05$.

As can be seen in Figure 4.5, the displacement rate, which is approximately analogous to the strain rate for this test case, greatly affects the results. Although the displacement rate of Figure 4.4 is still too high to really be defined as quasi-static loading, one can see from the kinetic energy that, initially, the loading is not really dynamic.

This is not the case when the displacement rate is increased. The cube is compressed much more quickly, such that the vertical struts fracture quite early on, which explains the difference in energy plots. One will notice too that the total mechanical energy eventually starts diverging slightly from the external work, likely because of contact interactions between the struts and the fractured elements. When the time step is insufficiently small and if the contact penalty is large, then collisions between the fractured elements could cause energy injection.

4.2. Case II: unit insideBCC subjected to an initial velocity front

The second test application features a unit insideBCC subjected to an initial velocity front at its top nodes. Essentially, the cube is subjected to an impulse load without needing to model an impactor. This test aims to investigate the fracture behaviour and energy balance under dynamic loading. The same unit cell is used as in Section 4.1, which can be seen in Figure 4.1. The simulation parameters for this test have been summarised in Table 4.2.

Table 4.2: Simulation parameters for case II: the impulse loading of a single insideBCC cell.

Parameter	Symbol [unit]	Value
Reference strut length	L_0 [m]	0.1
Young's modulus	E [MPa]	0.5
Strut radius	r [m]	0.001
Strut mass density	ρ [kg/m^3]	1000
DG interface penalty	β_p and β_t [-]	1000
Cohesive strength	σ_c [Pa]	9000
Fracture energy	G_c [N/m]	1
Mode-mixity parameter	α [-]	1
Initial velocity	v_0 [m/s]	-0.1
Contact penalty stiffness factor	R_{contact} [-]	10.0
Nº elements per strut	N [-]	10
Time step	Δt [s]	1.0e-6
Final time	t_{final} [s]	0.25

Most of the parameters in Table 4.2 are unchanged compared to the previous test. Instead of displacement loading, an initial velocity, v_0, is applied in this test. This initial velocity is applied to all top nodes of the cube, and it acts in the z-direction. Also, the time step Δt is a factor 10 smaller. The contact penalty factor is 10.0, but it has little effect on the results for $v_0 = 0.1$ m/s because as one shall see, there is no fracture in this run. In any case, using the material data and Equation 4.4, one finds that $k_p = 157.08$ N/m.

Boundary conditions wise, the bottom nodes are all still fixed just like in case I, but the top nodes do not have a prescribed displacement any more since this test is not displacement controlled. Instead, an initial velocity front is introduced as an initial condition to the solver.

A schematic of the loading in case II has been included as Figure 4.6. One can see that the bottom of the cube is constrained, but now there is an initial velocity front V_0 applied at the top of the cube.

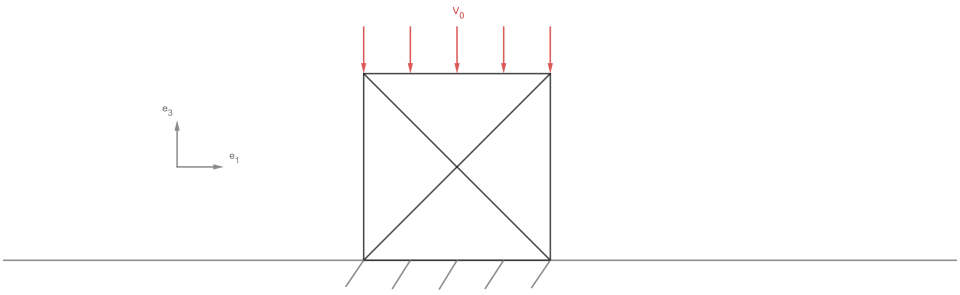
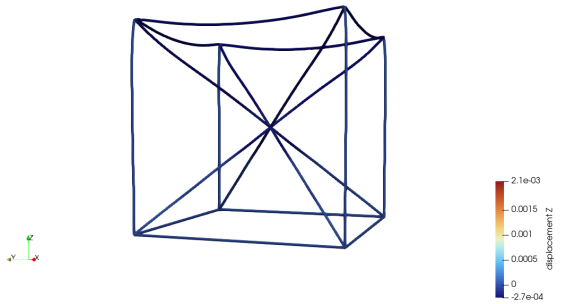


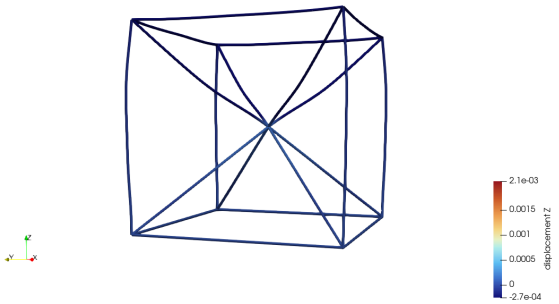
Figure 4.6: A schematic of the loading of test case II, made in GeoGebra.

4.2.1. Results

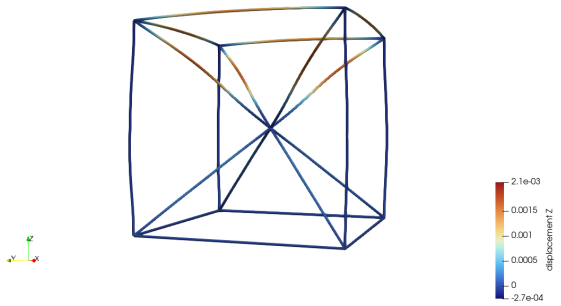
Four frames have been extracted from the simulation run and have been included in Figure 4.7.



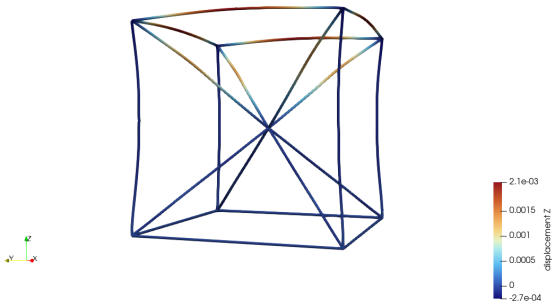
(a) Simulation frame for case II captured at approximately 0.063 s.



(b) Simulation frame for case II captured at approximately 0.125 s.



(c) Simulation frame for case II captured at approximately 0.188 s.



(d) Simulation frame for case II captured at approximately 0.25 s.

Figure 4.7: Four simulation frames for case II. Each frame corresponds to approximately 0.0004 s.

Figure 4.7d is the last frame of the simulation, so it also represents the final state of the system. One important thing of note is that the four frames do not do the simulation justice in the sense that they cannot capture the dynamic response properly. The initial velocity front causes the cube to vibrate. Because of the low initial velocity, the cube does not exhibit any fracture at all. Indeed, the initial velocity was too low to obtain any interesting results, so the same test case was run with an initial velocity $v_0 = -0.5$ m/s and $v_0 = -1.0$ m/s. Their final frames have been included below as Figure 4.8 and Figure 4.9.

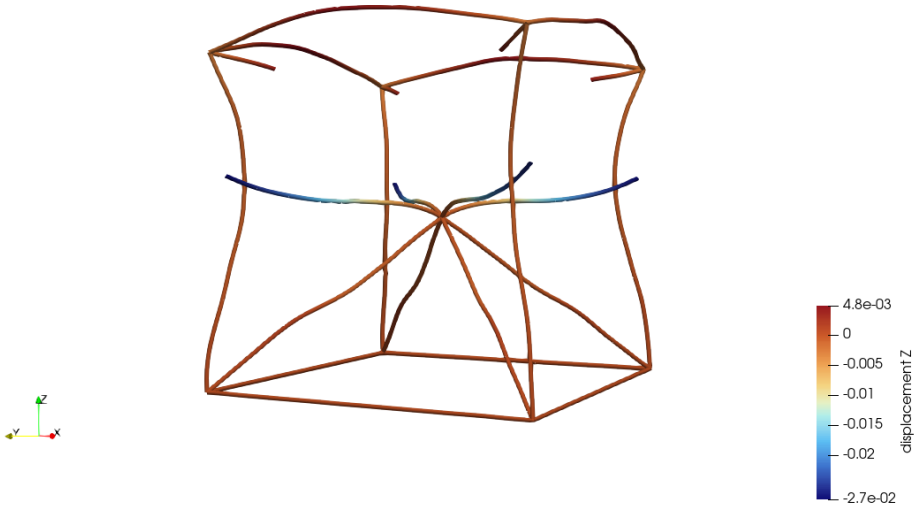


Figure 4.8: The final state of case II for $v_0 = -0.5$ m/s.



Figure 4.9: The final state of case II for $v_0 = -1.0$ m/s.

As can be seen from the figure above, the initial velocities for both runs were high enough this time to cause fracture. In the case of Figure 4.8, the initial velocity front had a large enough magnitude to cause fracture at the top diagonal struts, but not high enough to cause fracture in the vertical struts. For the run that produced Figure 4.9, the initial velocity front was strong enough to cause fracture in the vertical struts and the top diagonal struts, resulting in the final state seen in that figure. In both cases, the contact penalty came into play again because the fractured members can contact other parts of the structure. As such, for $v_0 = -0.5$ m/s, $R_{\text{contact}} = 50.0$, while for $v_0 = -1.0$ m/s, $R_{\text{contact}} = 100.0$. The corresponding penalty stiffness values can be found by scaling k_p for $v_0 = -0.1$ m/s by a factor 5 and 10, respectively.

The energy components for the three runs, $v_0 = -0.1$ m/s, $v_0 = -0.5$ m/s and $v_0 = -1.0$ m/s, have been plotted below as Figure 4.10, Figure 4.11 and Figure 4.12.

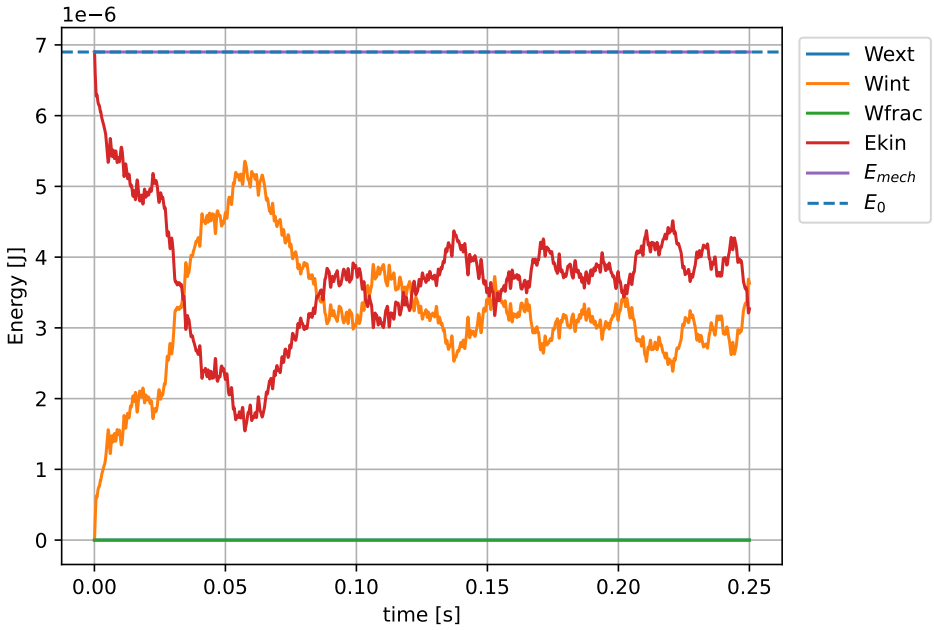


Figure 4.10: The energy balance of the system from case II for $v_0 = -0.1$ m/s.

Figure 4.10 shows two things. Firstly, there is energy balance, and indeed, conservation because of the lack of external forces. The mechanical energy of the system at all times is equal to the initial mechanical energy E_0. This is equal to the initial kinetic energy of the system caused by the initial velocity front. Secondly, E_{kin} and W_{int} look to be periodic in time, which lines up with the observed response of the cube, which vibrates because of the initial velocity front. The initial velocities being too low for fracture are a major reason why the system keeps vibrating since the lack of fracture means no energy dissipation.

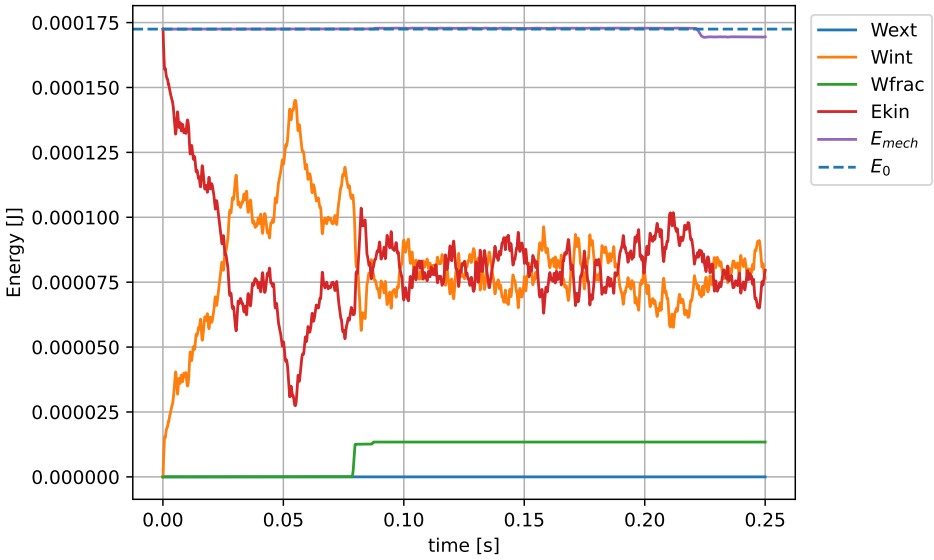


Figure 4.11: The energy balance of the system from case II for $v_0 = -0.5$ m/s.

In Figure 4.11, energy conservation is also maintained for the majority of the simulation except for the last time steps. v_0 is now large enough to cause fracture at the top diagonal struts, but the vertical struts have not failed and still vibrate, as can also be seen in the periodic waveforms of E_{kin} and W_{int}. The small dip in the mechanical energy of the system towards the end can be explained by the fact that the

top diagonal struts, which have broken off, now lie almost horizontally and actually do contact the vertical struts towards the end. This interaction may have lowered the kinetic energy of the system.

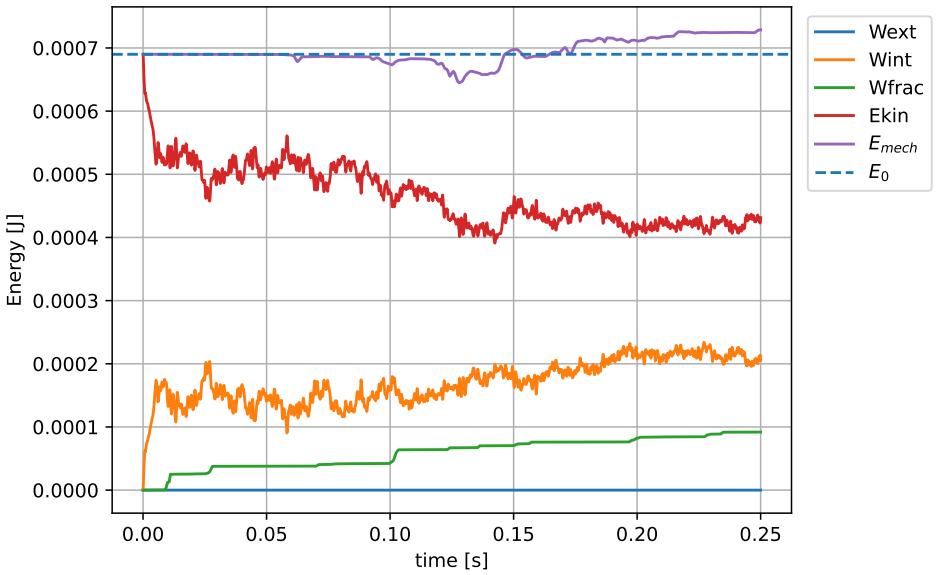


Figure 4.12: The energy balance of the system from case II for $v_0 = -1.0$ m/s.

For $v_0 = -1.0$ m/s, Figure 4.12 shows that the kinetic and internal energy are aperiodic, if the high frequency responses are discounted. This also lines up with the observed results in that v_0 is sufficiently high to cause fracture in the vertical struts and the top diagonal struts close to the start of the simulation. Secondary impact events after this initial fracture causes the broken struts to further fragment. One can see that the mechanical energy of the system stays roughly constant throughout, as one would expect for a conservative system.

However, there are two points, one after $t = 0.10$ s and the other after $t = 0.17$ s, where the mechanical energy starts diverging from E_0. It also happens to be the case that at and during these points, there are many contact events between the fractured pieces of the struts. The temporary dip in mechanical energy can be explained by the contact work phenomena explained in Subsection 3.3.4. Because the contact events involve smaller fragments contacting with intact struts, and because the time step has not been changed from $\Delta t = 1 \cdot 10^{-6}$ s while both the impact velocity and R_{contact} were increased, this could have led to a larger contact work contribution. Since this contribution is not tracked, it could appear as a dip in the mechanical energy.

The rise in the mechanical energy, however, may be more indicative of energy injection. One can see from Figure 4.12 that the mechanical energy after $t = 0.17$ s suddenly jumps up, but stays roughly constant afterwards. Similarly, the kinetic energy at around $t = 0.17$ s increases while it was still decreasing before this point. The internal energy has an increasing trend, so one would expect that the kinetic energy should decrease to maintain energy balance. However, if kinetic energy were to be injected into the system somehow at sudden points in time, then one would observe jumps in the kinetic energy but outside of those jumps, the kinetic energy would not continuously grow. This could lead to the case observed in Figure 4.12 where the mechanical energy increases suddenly, but it does not grow continuously.

The question is how contact could cause energy injection. One possible explanation is that the contact force and the associated impulse is too large to maintain conservation of energy. To try and explain this, consider a simplified example of a ball colliding with a wall, Figure 4.13.

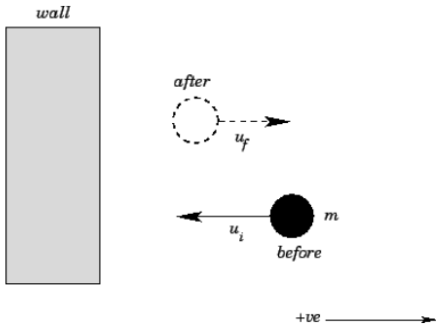


Figure 4.13: A schematic of a ball colliding with a wall, from the lecture notes by Richard Fitzpatrick (2006) on classical mechanics [105].

In the figure, the ball has a mass m, and it moves towards the wall at an initial velocity u_i. After impact, it has a final velocity of u_f in the opposite direction. Now, let F be the average contact force over the duration of the impact. Using the principle of linear momentum and impulse Equation 4.6, the final velocity u_f can be calculated.

$$\boldsymbol{L}_f - \boldsymbol{L}_i = \boldsymbol{I} \tag{4.6}$$

In Equation 4.6, $\boldsymbol{L}_f$ is the final linear momentum, $\boldsymbol{L}_i$ is the initial linear momentum, and $\boldsymbol{I}$ is the linear impulse. Applied to Figure 4.13, this equation results in $mu_f - mu_i = Fdt$, where dt is the impulse duration. This expression can be rearranged to solve for u_f: $u_f = u_i + \frac{Fdt}{m}$.

However, to maintain conservation of energy, there is an upper limit on the average contact force F imposed by the system's kinetic energy. The constraint on the kinetic energy is expressed as follows:

$$\frac{1}{2}mu_f^2 - \frac{1}{2}mu_i^2 \leq 0 \tag{4.7}$$

Note that Equation 4.7 is not an equality because the collision need not be fully elastic. Now, when $u_f = u_i + \frac{Fdt}{m}$ is substituted into Equation 4.7, one obtains the condition below.

$$\frac{1}{2}m\left(\frac{Fdt}{m} + u_i\right)^2 - \frac{1}{2}mu_i^2 \leq 0 \tag{4.8}$$

When Equation 4.8 is solved for F, one obtains $-2u_i\frac{m}{dt} \leq F \leq 0$. Using the sign convention used in Figure 4.13, where u_i is negative, then the restriction on F becomes $0 \leq F \leq -2u_i\frac{m}{dt}$. If $\|F\| > \|-2u_i\frac{m}{dt}\|$, then the collision would add kinetic energy to the system. This means that if F is freely chosen, then there is a real possibility that energy is injected into the system.

The way this ties into the contact model is that the penalty approach essentially prescribes the contact force via the contact penalty stiffness. Effectively, the contact force is freely chosen in the penalty approach. If the contact penalty is large, and if the interpenetration is large as a result of a large time step, for instance, then the contact force over the duration of contact will also be large, leading to a large average contact force. If a similar condition on the contact force exists for beam contact problems to not add energy to the system, then it is possible that in the case of Figure 4.12, the average contact force over the contact duration violates this condition, resulting in the injection of energy into the system. Of course, one must keep in mind that the example from Figure 4.13 is a collision of a ball with a wall. Equating this example to the case of contacting beam elements would be a major oversimplification, but the underlying physics of impact indicates that the contact forces cannot be too large lest the change in momentum of the contacting elements increases their kinetic energy.

4.3. Case III: collision of two insideBCC unit cells

The third test case concerns one insideBCC unit cell colliding with another insideBCC unit cell. In a way, case III is essentially case II, but this time, the impactor is explicitly modelled using another insideBCC

cell. The goal of this test is to investigate the effect of contact on the energy balance during collision. Both unit cells are insideBCC unit cells like seen in Figure 4.1, however, the set-up of the collision test deserves its own figure given that it is slightly more involved than the previous cases, see Figure 4.14.

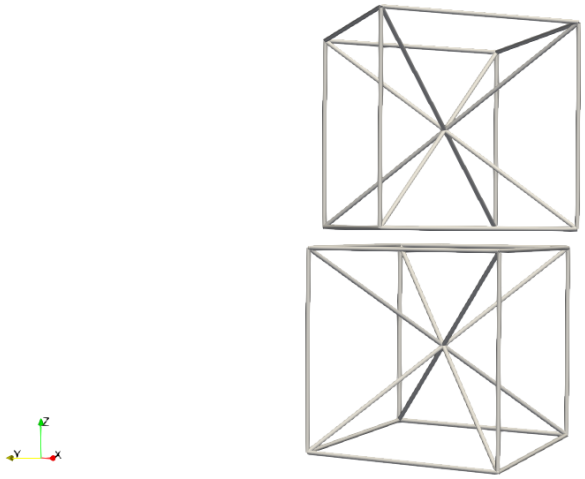


Figure 4.14: The set-up for case III. The impactor unit cell is rotated 45 degrees around the z-axis compared to the stationary unit cell, and it moves downwards with 3.0 m/s.

The simulation parameters used in this test run have been summarised in Table 4.3.

Table 4.3: Simulation parameters for case III: the collision of two insideBCC cells.

Parameter	Symbol [unit]	Value
Reference strut length	L_0 [m]	0.1
Young's modulus	E [MPa]	430
Strut radius	r [m]	0.001
Strut mass density	ρ [kg/m^3]	1100
DG interface penalty	β_p and β_t [-]	1000
Cohesive strength	σ_c [Pa]	11.0e6
Fracture energy	G_c [N/m]	2
Mode-mixity parameter	α [-]	1
Initial velocity	v_0 [m/s]	-3.0
Contact penalty stiffness factor	R_{contact} [-]	10.0
Nº elements per strut	N [-]	5
Time step	Δt [s]	1.0e-8
Final time	t_{final} [s]	0.025

The biggest change in parameters compared to case II is the fact that a different material is used. Case I and case II used a very compliant material with $E = 0.5$ MPa. The problem is that such a material does not really exist in real life, which also means that there is no fracture data on it. As such, the σ_c and G_c parameters used in case I and case II have no physical basis. Because the main computational experiment of impact in a lattice structure from Chapter 5 does use real materials, case III uses one of the materials which can be used for said experiment. In this case, TMPTA.

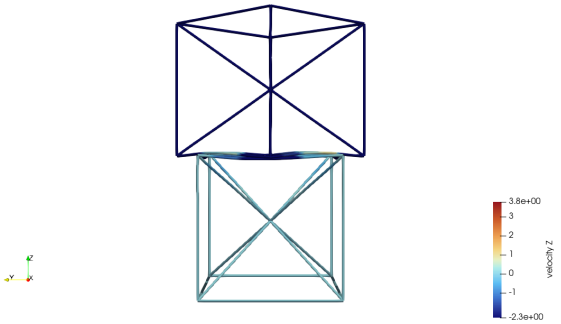
Because TMPTA has a much higher stiffness, it necessitates running this simulation case with a much smaller time step size. To save on computational time, the number of elements per strut was also reduced to 5. The contact penalty factor was set to 10.0 as an initial guess of for the contact penalty.

Using the material data provided in Table 4.3 and using Equation 4.4, the penalty contact stiffness amounts to $k_p = 135088$ N/m. After presenting the results for $R_{\text{contact}} = 10.0$ and $\Delta t = 1 \cdot 10^{-8}$ s, the results of a simple sensitivity study for the contact penalty factor R_{contact} and a convergence study for Δt are presented to illustrate the influence of both R_{contact} and Δt on the energy response of the problem.

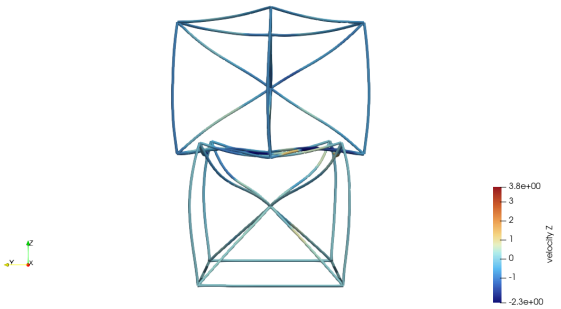
Just as in case I and case II, the bottom nodes of the fixed cell are fixed. v_0 is applied to the entirety of the impactor cube, and it acts in the z-direction.

4.3.1. Results

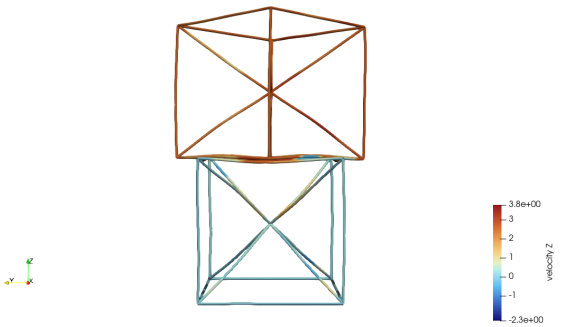
Four frames have been extracted the simulation run and have been included below as Figure 4.15.



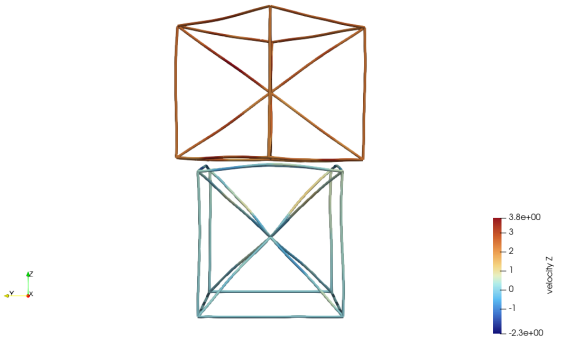
(a) Simulation frame for case III captured at approximately 0.00375 s.



(b) Simulation frame for case III captured at approximately 0.01 s.



(c) Simulation frame for case III captured at approximately 0.02 s.



(d) Simulation frame for case III captured at approximately 0.02375 s.

Figure 4.15: Four simulation frames for case III. Each frame corresponds to approximately 4.0e-5 s.

The final state of the system has also been captured, and included as Figure 4.16.

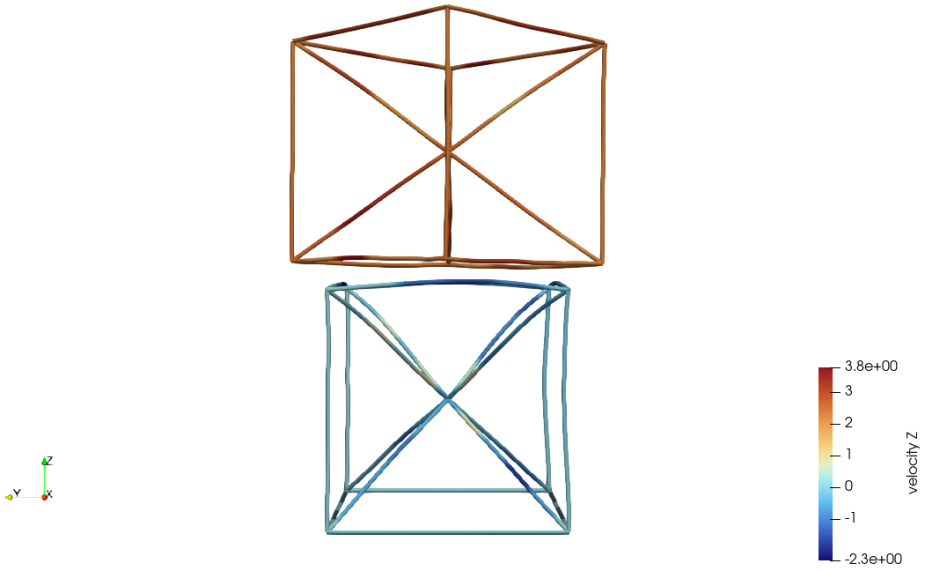


Figure 4.16: The final state for case III.

One thing which immediately stands out from Figure 4.15, is that there is no fracture. The initial velocity of the falling cube is not high enough to cause fracture. This is an inevitable result of choosing a much stiffer and stronger material. For this initial velocity, the deformations of the cubes are not large enough to cause stresses that can initiate fracture in the struts. However, the much more compliant material from the two cases before would actually cause the cubes to disintegrate on collision for the same initial velocity, which also would not yield much meaningful information. This experiment must therefore be rerun with a higher initial velocity to achieve fracture.

The energy components for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 10.0$ have been plotted below as Figure 4.17.

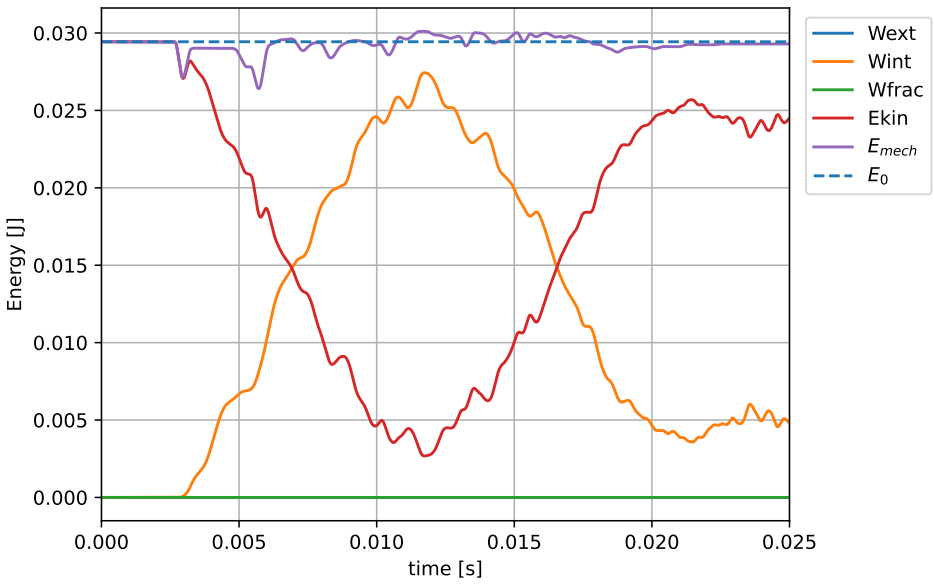


Figure 4.17: The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 10.0$.

Figure 4.17 confirms that there is no fracture in this simulation run. Furthermore, one can see that the mechanical energy of the whole system, that is both falling and fixed cubes, is roughly constant and equal to the initial energy. This behaviour is as expected since there are no external forces to this system acting on the system. That said, during contact (roughly $0.00375 \leq t \leq 0.022$ s), there are some

fluctuations in the mechanical energy. It stands to reason that the contact forces during collision are the cause of this.

A further indication of this is the fact that post-contact, the mechanical energy of the system stays constant again, meaning no new energy is injected into the system after the collision, and it returns to the same value as the initial energy. This seems like a similar case to the temporary dip in mechanical energy observed in Figure 4.12 where the phenomenon could be explained as the result of a large time step size and high contact penalty stiffness. Because the contact work contribution is not tracked, this could cause fluctuations in the mechanical energy.

As may have been the case with Figure 4.12, a combination of large time step size and contact penalty stiffness can cause two contacting members to gain a large change in velocity due to the contact forces acting on them. If the time step size is not small enough, then the initial velocity of the top cube can cause large initial interpenetration, which, when combined with a large contact penalty stiffness, can cause a large contact force on the contacting elements. One way to remedy this, may be to keep on decreasing both the time step size and the contact penalty stiffness until the mechanical energy no longer fluctuates during contact. To verify this, a sensitivity study on the contact penalty stiffness was performed, and a convergence study for Δt was conducted.

4.3.2. Sensitivity study of R_{contact}

To investigate the effect of the penalty stiffness on the energy response, a sensitivity study was conducted. The value of R_{contact} was varied from 2.5 to 20.0 and a simulation was run for each of the R_{contact} values in $[2.5, 5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0]$ except for $R_{\text{contact}} = 10.0$ since this is the same case as Figure 4.17. The actual penalty contact stiffness values k_p can be calculated by scaling k_p for $R_{\text{contact}} = 10.0$ for the other values of R_{contact}. For each simulation, an error measure, e_S, was computed and plotted against the contact penalty factor. The error measure is defined as Equation 4.9.

$$e_S = \frac{1}{t_{\text{final}} - t_0} \int_{t_0}^{t_{\text{final}}} \sqrt{\left(\frac{E_{mech}(t)}{E_0} - 1\right)^2}\, dt \tag{4.9}$$

This error measure expresses the cumulative root squared relative energy error. The instantaneous relative energy error $\sqrt{\left(\frac{E_{mech}(t)}{E_0} - 1\right)^2}$ is integrated over time and then normalised by the time span so that this error measure is dimensionless. The reason for integrating the relative energy error over time is to take the time history of the energy into account. Just because the relative energy error at $t = t_{\text{final}}$ is low, does not mean the cumulative relative energy error is low. A case in point is Figure 4.17: the relative energy error near the end is very low, but one can see that the time history of E_{mech} shows noticeable fluctuations. The square root of the square of the relative errors was taken to make all relative errors positive so that errors of opposite signs cannot balance each other out. This way, relative errors will strictly add to the cumulative error measure.

The error plot has been included as Figure 4.18.

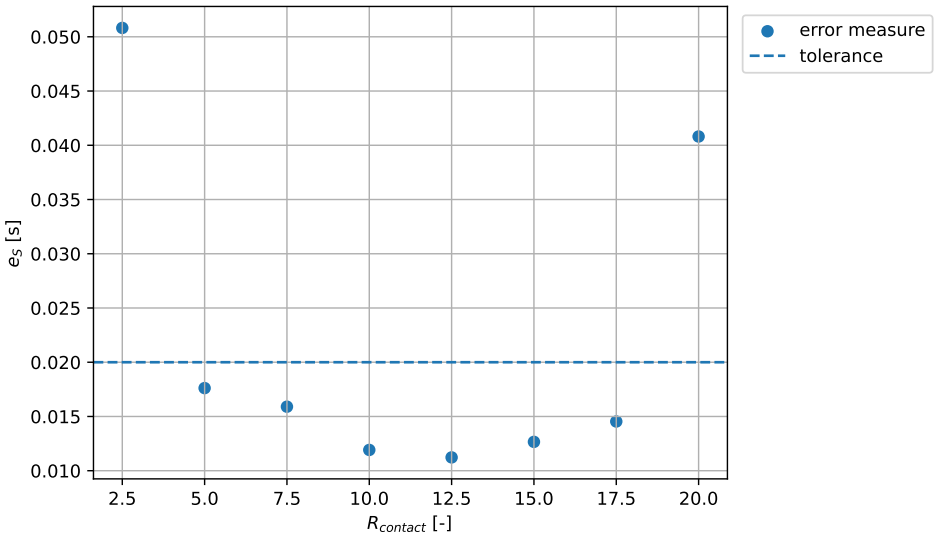


Figure 4.18: The error plot for the mechanical energy from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} \in [2.5; 20.0]$. e_S is the cumulative relative error of the mechanical energy.

Figure 4.18 illustrates clearly that the energy response depends on the penalty stiffness. One will also notice that the error plot is not monotonic. Increasing R_{contact} to 12.5 lowers the cumulative relative energy error, but increasing it further causes e_S to rise again. This is in line with what was discussed in Subsection 4.2.1: increasing R_{contact} while keeping Δt fixed may lead to energy injection. On the other hand, lowering R_{contact} from the initial value of 10.0 also increases e_S. Lowering the contact penalty causes more interpenetration, which may result in increased contact work, as discussed in Subsection 4.2.1.

The tolerance line in Figure 4.18 is the tolerance for e_S. It is equal to $e_{tol} = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \epsilon_{tol}(t) dt$, where $\epsilon_{tol}(t)$ is the instantaneous relative error tolerance. A constant value of 2% was chosen for this instantaneous tolerance. Because the instantaneous tolerance is constant, the e_{tol} is simply 2% multiplied by $t_f - t_0$, divided by $t_f - t_0$, resulting in an error tolerance of 0.02. According to this tolerance, the results for $5.0 \leq R_{\text{contact}} \leq 17.5$ can already be considered converged. It is only when the contact penalty factor falls outside this range that e_S is greater than the tolerance. To show why this is the case, the energy responses for $R_{\text{contact}} = 20.0$ and $R_{\text{contact}} = 2.5$ are shown below. The energy plots for the other values of R_{contact} can be found in Section A.1.

Figure 4.19 illustrates the energy balance for when the contact penalty factor is 20.0.

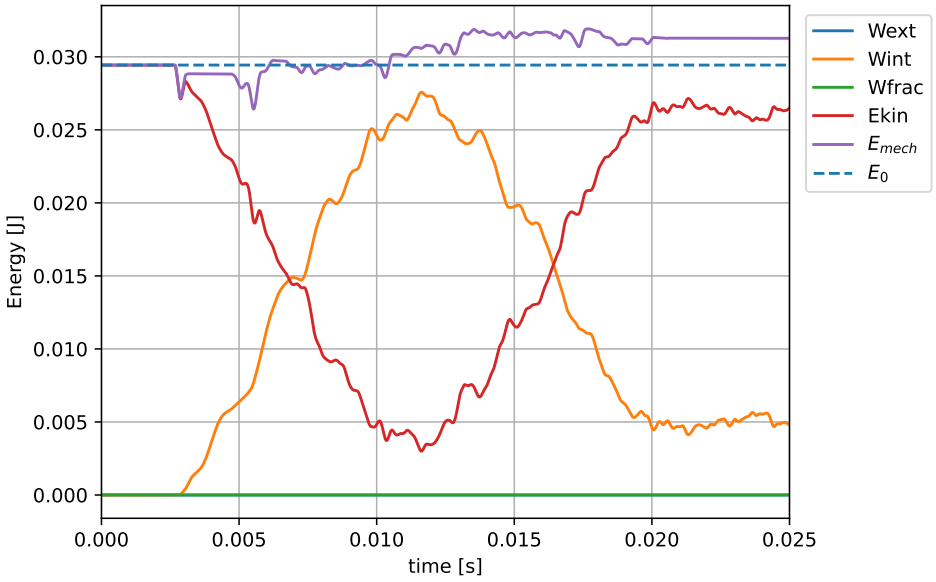


Figure 4.19: The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 20.0$.

One can see that energy is injected into the system during contact, but after contact, the mechanical energy maintains balance again. This seems similar to what was observed in Figure 4.12 for $t > 0.17$ s where the contact penalty stiffness may have been too high to maintain energy balance. Figure 4.19 along with Figure 4.18 demonstrate that increasing the penalty contact stiffness while keeping the same time step size can cause numerical errors.

Figure 4.20 is the energy balance for $R_{\text{contact}} = 2.5$.

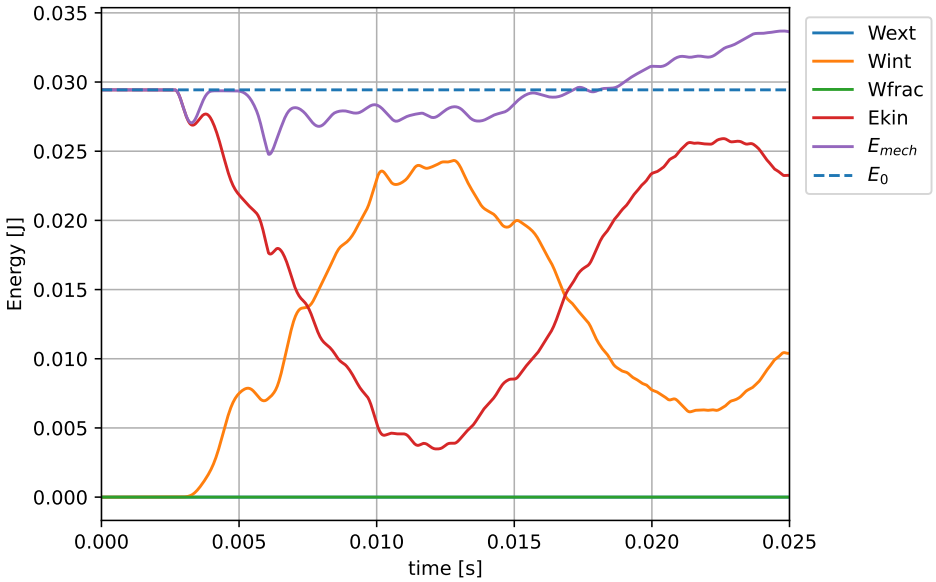


Figure 4.20: The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 2.5$.

What stands out about the energy response is that the mechanical energy past $t > 0.017$ s keeps rising. The same trend can be observed in the internal energy past this point. The reason for this turns out to be missed contact events, as can be seen in Figure 4.21.

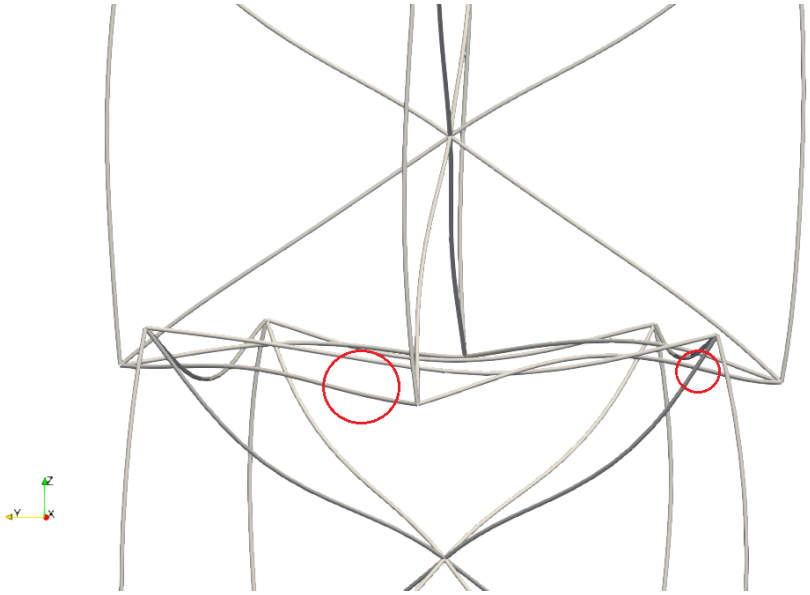


Figure 4.21: Missed contact in case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 2.5$.

When the penalty contact factor becomes as low as 2.5, the collision between the two cubes actually results in some missed contact events. This results in some struts of the top cube phasing through the struts of the bottom cube. Two instances of missed contact have been highlighted in Figure 4.21. When the cubes start moving apart again, the entangled struts pull some of the struts of the bottom cube up with the top cube, resulting in the increasing trend of the internal energy observed in Figure 4.20. This all goes to show that there is also a lower limit on the penalty contact stiffness for a given time step size. Going below this limit may not only cause excessive interpenetration, but may even cause missed contact events.

In all, the sensitivity study on the contact penalty stiffness demonstrates that the value of R_{contact} definitely affects the results. At a given time step size Δt, there is a "safe" range of R_{contact} for which the energy error is acceptable. However, going outside this range, either above or below, may cause large errors in the energy time history either as a result of energy injection when the contact penalty is too high, or because of missed contact events when the contact penalty is too low. The implication for tuning the penalty stiffness is that there is no clear rule to follow: one cannot keep on increasing the contact penalty since this may eventually actually increase the energy error, but one also cannot keep on decreasing the contact penalty for the same reason.

Therefore, simply decreasing the contact penalty stiffness will not always be the solution. This is in contrast to a convergence study with Δt where decreasing the time step will normally lead to lower error and increasing the time step size does the opposite. A practical method for tuning the contact penalty stiffness would be to find the lowest penalty value for which no contact events are missed and for which whatever error measure is chosen, such as e_S in this study, is below the tolerance. If the error measure cannot be kept below the tolerance, one will have no choice but to reduce Δt.

4.3.3. Convergence study with Δt

The influence of the time step size Δt on the energy response was also analysed. The original intention was to conduct a convergence study where the time step Δt would be refined from the original value of $\Delta t = 1 \cdot 10^{-8}$ s to $\Delta t = 1 \cdot 10^{-9}$ ns with an intermediate time step size of $\Delta t = 5 \cdot 10^{-9}$ ns. In other words, the range of Δt for which the cumulative relative energy error, e_S, would be calculated is $\Delta t \in [1 \cdot 10^{-8}, 5 \cdot 10^{-9}, 1 \cdot 10^{-9}]$ s. The resulting error plot is given by Figure 4.22.

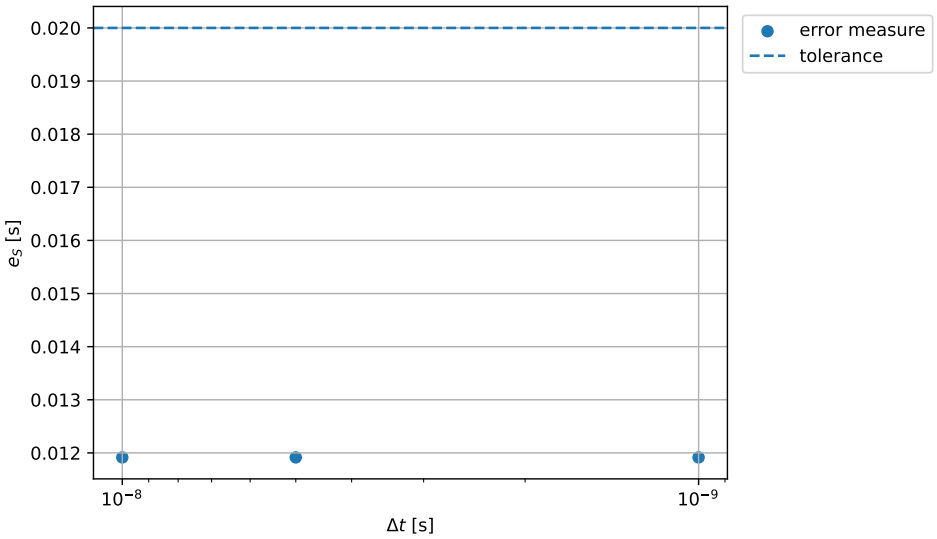


Figure 4.22: The error plot for the mechanical energy from case III for $v_0 = -3.0$ m/s and $\Delta t \in [10, 5, 1]$ ns. e_S is the cumulative relative error of the mechanical energy. Logarithmic scaling is used on the x-axis. Note that the x-axis has been inverted.

Figure 4.22 reveals something which should already have been apparent from Figure 4.18: if an instantaneous relative error tolerance of 2% is chosen, which also results in a cumulative relative error tolerance of 0.02 when integrated over the time span $t \in [0; 0.025]$ s and subsequently normalised by the same time duration, then $\Delta t = 1 \cdot 10^{-8}$ s was already enough to satisfy the error tolerance. As Figure 4.22 shows, further decreasing Δt has no observable effect on the error measure, so the solution had already converged for the original time step size used in Figure 4.17.

In the end, it does not seem that for $R_{\text{contact}} = 10.0$ and $v_0 = -3.0$ m/s, changing Δt has much effect on the energy response. One can find the energy response for the smaller values of Δt in Section A.2. Rather, where the time step may start playing a more prominent role can be inferred from Figure 4.18. Subsection 4.3.2 concluded that for a given time step Δt, there appears to be a range of values of R_{contact} for which e_S is acceptable. The bounds of this range seem to be determined on the low end by the minimum contact penalty value to prevent missed contact, and on the high end by the maximum contact penalty for which collisions inject energy into the system.

It is likely that decreasing Δt increases the range of safe R_{contact} values. The lower bound can be pushed lower because a smaller time step helps prevent missed contact events by virtue of there being less of a time window for elements to phase through each other. For a similar reason, the upper bound on R_{contact} could be pushed higher because a smaller time step limits the maximum interpenetration possible in one time step, which also directly limits the maximum change in contact force possible. This means the impulse due to contact is lower, which helps prevent energy injection during contact.

However, to prove this hypothesis, one would need to conduct the same sensitivity study in Subsection 4.3.2 for different values of Δt and find the "safe" range of R_{contact} values. If the hypothesis holds, then the range of R_{contact} for which e_S is acceptable increases as Δt is lowered from $\Delta t = 1 \cdot 10^{-8}$ s. This makes for a key recommendation for potential future work.

4.4. Case IIIb: collision of two insideBCC unit cells at higher velocity

The results of test case III indicate that the time step size and the contact penalty parameter affect the energy balance. However, because the initial velocity $v_0 = -3.0$ m/s was too low to result in any interesting effects as seen in Figure 4.16, it was decided to run case III with a much higher initial velocity $v_0 = -50.0$ m/s, which is more comparable to the initial velocities used in Chapter 5. This test case, dubbed case IIIb, is much more likely to show fracture behaviour compared to case III. Because of the higher initial velocity, the penalty parameter and the time step size were retuned for this case.

The set-up is still mostly the same as case III, with the initial state being the exact same as Figure 4.14. The simulation parameters used in this case are presented in Table 4.4.

Table 4.4: Simulation parameters for case IIIb: the collision of two insideBCC cells at higher velocity.

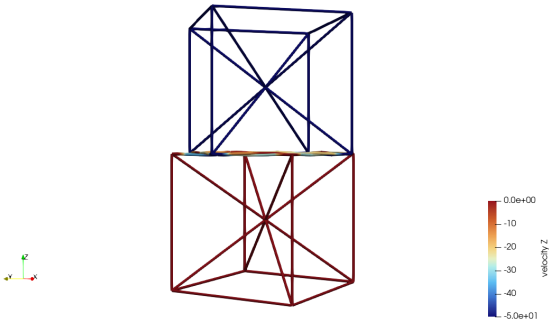
Parameter	Symbol [unit]	Value
Reference strut length	L_0 [m]	0.1
Young's modulus	E [MPa]	430
Strut radius	r [m]	0.001
Strut mass density	ρ [kg/m^3]	1100
DG interface penalty	β_p and β_t [-]	1000
Cohesive strength	σ_c [Pa]	11.0e6
Fracture energy	G_c [N/m]	2
Mode-mixity parameter	α [-]	1
Initial velocity	v_0 [m/s]	-50.0
Contact penalty stiffness factor	R_{contact} [-]	1000.0
Nº elements per strut	N [-]	5
Time step	Δt [s]	1.0e-10
Final time	t_{final} [s]	0.01

One may notice that compared to case III (see Table 4.3), aside from v_0 being much larger in magnitude, the contact penalty factor R_{contact} is much larger and the time step size Δt is far smaller. Both were necessary changes to (1) not miss the impact event entirely, and (2) not cause excessive energy injection during impact. t_{final} was reduced to 0.01 s since the impact event itself will not take as long because the initial velocity is much higher.

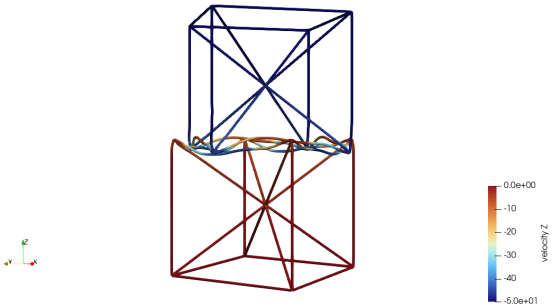
The boundary conditions have not changed, so the bottom nodes of the stationary cube have been fixed, and v_0 is applied to all nodes of the impactor cube, acting in the z-direction.

4.4.1. Results

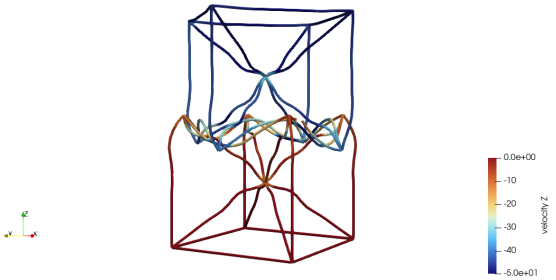
Just like with case III, four frames have been extracted from the simulation. These can be seen in Figure 4.23.



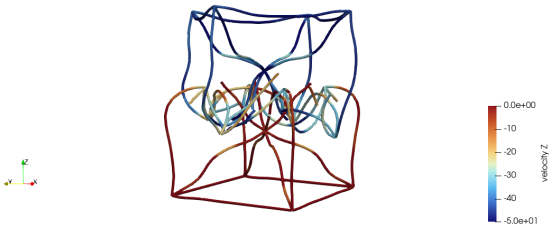
(a) Simulation frame for case IIIb captured at approximately 0.0002 s.



(b) Simulation frame for case IIIb captured at approximately 0.0004 s.



(c) Simulation frame for case IIIb captured at approximately 0.0008 s.



(d) Simulation frame for case IIIb captured at approximately 0.0016 s.

Figure 4.23: Four simulation frames for case IIIb. Each frame corresponds to approximately 4.0e-7 s.

Due to time constraints, the final state of the system at $t_{\text{final}} = 0.01$ s was not captured at this time. The simulation of case III already took close to 4 days on an HP ZBook Power G11 Mobile Workstation laptop. Case IIIb was run with a time step a factor 100 smaller compared to case III, but the final time is only a factor 10 smaller since the initial impact velocity is not so high compared to case III that it warrants cutting the simulation time down by a factor 100. As such, running the simulation until its actual conclusion would take far more than four days. Instead, the state of the system at $t \approx 0.00240$ s, which is almost four days' worth of simulation time, is shown in Figure 4.24.

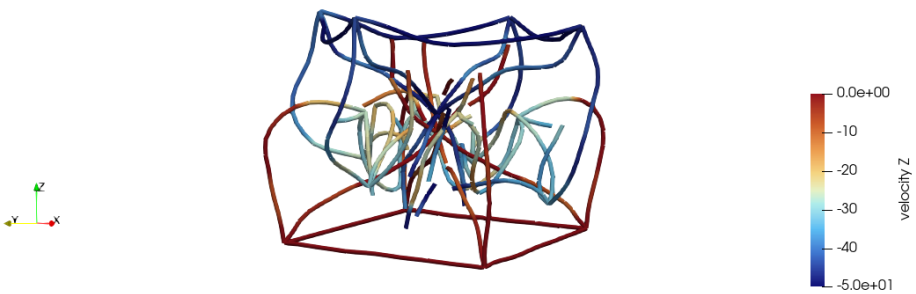


Figure 4.24: The state for case IIIb at $t \approx 0.00240$ s.

Now, one can observe from Figure 4.23 that under an impact velocity of 50.0 m/s, the colliding cells do exhibit fracture in the struts. The frame shown in Figure 4.23a is the state of the system during the initial impact. At this point, there is no fracture yet as the bottom cube catches the top cube and the top struts start deforming during contact. From here, the struts in contact deform in bending, which can also be seen in Figure 4.23b. The horizontal struts in contact keep bending more and more until they start fracturing, as can be observed in Figure 4.23c. After this, the top cube hits the upper diagonal struts of the bottom cube, which also fracture, as shown in Figure 4.23d. The last capture frame, Figure 4.24 at $t \approx 0.00240$ s, still depicts the remnants of the impactor cube falling downwards after it has hit the lower diagonal struts of the bottom cube.

A plot of the energy components of the system for $v_0 = -50.0$ m/s and $R_{\text{contact}} = 1000.0$ has been included as Figure 4.25.

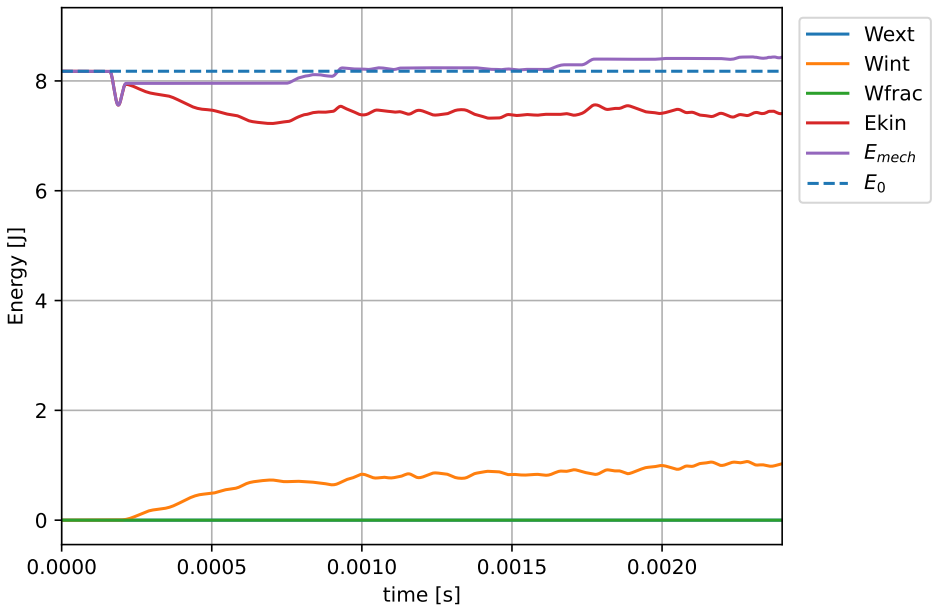


Figure 4.25: The energy balance of the system from case IIIb for $v_0 = -50.0$ m/s and $R_{\text{contact}} = 1000.0$.

Ironically, because the initial velocity is so high, the kinetic energy is so much higher than the fracture energy that W_{frac} cannot be observed from Figure 4.25. Therefore, the plot has also been generated with logarithmic scaling on the y-axis. This way, W_{frac} can be distinguished and can indeed be verified to be non-zero. This semi-logarithmic plot can be seen in Figure 4.26.

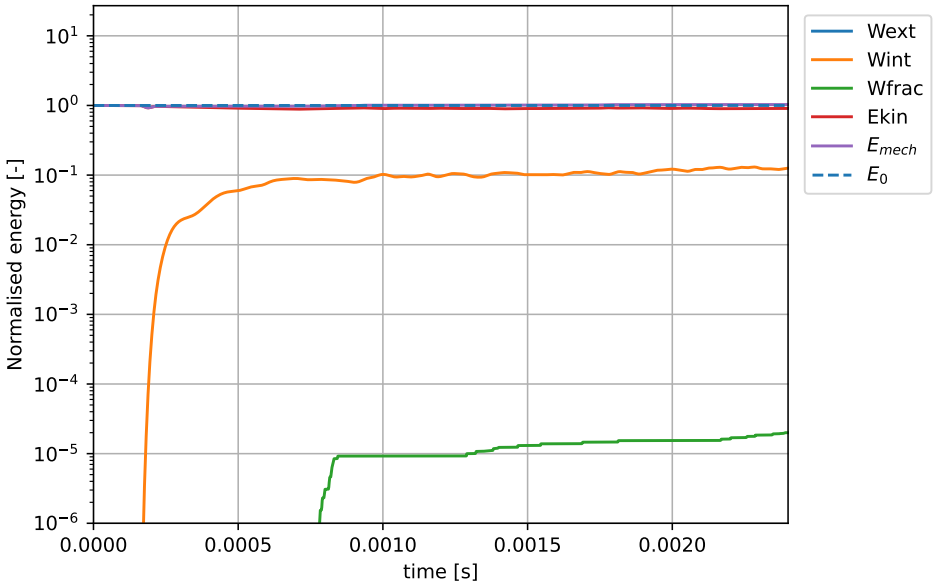


Figure 4.26: A semi-logarithmic plot of the energy balance of the system from case IIIb for $v_0 = -50.0$ m/s and $R_{\text{contact}} = 1000.0$. The y-axis has a logarithmic scale so that W_{frac} is better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable.

From Figure 4.26, one can observe that W_{frac} suddenly increases as the struts start fracturing. It should be noted that the lower limit on the y-axis was set at 10^{-6} J so that the plot was still readable as a whole. Because W_{int} and W_{frac} are 0.0 at the beginning, the y-axis would extend to 10^{-22} and lower, completely skewing the entire plot. To prevent this, a more reasonable lower limit was chosen.

Looking back at Figure 4.25, the energy of the system is generally in line with one's expectations. Although there are two slight abnormalities with the energy plot, one can see that the system's energy is roughly balanced. Post-impact, the mechanical energy E_{mech} of the system is approximately between 8.074 J and 8.434 J, while the initial energy E_0 is about 8.176 J. This gives a relative error of roughly 1.2% – 3.2%. As surmised from Section 4.3, an even lower relative error should be possible by further lowering the time step and tuning the contact penalty.

The first abnormality with Figure 4.25 is the dip in E_{mech} around $t = 0.0002$ s. This is around the time the two cubes collide with each other. The raw energy component data reveal that the kinetic energy decreases due to the collision, but the internal energy does not rise in equal amount during the same time. Something similar can be observed early in the energy response in Figure 4.17 too. There, a temporary dip in the mechanical energy is also observed, which eventually evens out. It is possible that in Figure 4.25, this deviation is also caused by the work done by the contact forces. Since this contribution to the energy is not tracked, it could appear as a temporary fluctuation of the mechanical energy.

The second abnormality observed in the plot is the fact that the E_{mech} makes small jumps at around $t = 0.0009$ s and $t = 0.00175$ s. One can draw parallels to Figure 4.12, and indeed in this case, it does seem like contact events after the initial collision could explain the deviations. At around $t = 0.0009$ s, the top cube's struts start making contact with the upper diagonal struts of the bottom cube. Meanwhile, after $t = 0.00175$ s, there are a number of fragments caused by the fracture of some of the struts, which could have collided with the remaining struts. As Subsection 4.3.2 has shown, if the time step size is not small enough or if the contact penalty parameter is not tuned just right, then contact events between the beam elements can inject energy into the system. For case IIIb, the time step and contact penalty parameter were tuned for the initial collision between the two cubes, but it is possible that secondary impact after this may not be correctly captured.

Of course, $\Delta t = 1 \cdot 10^{-10}$ s is already a small time step, and lowering it would slow down the simulation

even more than it already has. One must also consider how grave these abnormalities are when a larger and more practical lattice structure is tested, rather than just two unit cells. It is possible that the aberrations compound, but it is also possible that the deviations become smaller relative to the total energy if the impactor is smaller than the stationary lattice structure.

4.5. Comments on the test applications

To summarise the results of the test applications: three test cases were investigated. In case I, an insideBCC cell is gradually compressed. The results indicate that energy balance is maintained even when fracture starts occurring. One can observe a clear dependence of the response on the displacement rate.

In case II, an insideBCC cell is subjected to an initial velocity front at its top nodes. The resulting energy plots maintain balance quite well, although for the highest velocity run, it would seem like the contact interactions between the fragments caused the mechanical energy to slightly diverge over time.

Lastly, for case III, two insideBCC cells collide. This case was run twice, the second simulation run being named case IIIb to distinguish it from the first. The results of case III suggest that the contact penalty and the time step size generally must be tuned in a convergence study to properly balance energy. However, as long as the contact penalty value is not too low to cause missed contact events, the aberrations in the energy balance mostly come in the form of sudden jumps in the energy or temporary dips. In time, the mechanical energy becomes constant again after contact which implies the energy is stable after contact.

Because of the hypothesised dependence of the response on the contact penalty stiffness and Δt, a sensitivity study was performed on the contact penalty stiffness and a convergence study was conducted on Δt. The contact penalty sensitivity study indicated that there is at a given Δt, there is a range of values of contact penalty stiffness for which the cumulative relative energy error stays below the tolerance. The convergence study showed that the original value of Δt at which case III was run already led to convergence for a fixed contact penalty stiffness and initial impact velocity, and reducing Δt further did not lower the error appreciably. However, it is possible that reducing Δt increases the range of contact penalty values for which the cumulative relative energy error stays below the tolerance. To verify this, though, it is recommended for future work to perform the contact penalty sensitivity study for different values of Δt.

The results of case IIIb have been obtained after tuning the time step and the contact penalty parameter to try and balance the system energy. Unlike case III, case IIIb had a higher impact velocity, resulting in fracture of the struts being observed. The energy balance is still not perfect, but it shows that the same approach of lowering Δt and tuning R_{contact} works for a much higher initial velocity.

From the results discussed above, one recurring theme is that contact interactions can seriously affect the calculated energy components in a system. Case III and IIIb demonstrate this in particular. However, it can also be said that the energy plots are more sensitive to parameters such as the contact penalty and the time step size. Energy components like the kinetic energy and the internal energy are functions of the velocities squared, and the strains and curvatures squared, respectively. Any error in those variables can be magnified in the energy calculation. This means that convergence studies must be conducted using the energy balance as a measure of convergence rather than the displacement response.

5

Results and discussion

Subtask 3

Apply the unified beam-contact-fracture framework to a large-scale numerical model of impact in a truss metamaterial and evaluate the effect of self-contact on the energy absorption.

The test cases in Chapter 4 dealt with small-scale problems not larger than two unit cells in all. This chapter is dedicated to the main numerical experiment, a large-scale metamaterial impact simulation run on a high-performance computing cluster. First, the set-up is described and the parameters that go into the simulations are discussed. After that, the results of the simulations are presented and discussed. These results include the trajectory data used for interpreting the qualitative impact response of each specimen, and the energy data employed to analyse the quantitative impact response of each specimen.

5.1. Simulation set-up

The simulations are run for multiple test cases. Each case is a specific combination of initial impactor velocity and specimen geometry.

Before presenting the results, the set-up of the simulation and the input parameters that go into the computational model are shown and discussed. The inputs are categorised into geometric, material and simulation settings. Because there are multiple combinations of material and topology possible for each test case, the material properties of the three possible choices of material are shown below in Table 5.1.

Table 5.1: The properties of the three possible material choices.

Material	Name	ρ [kg/m^3]	E [Pa]	σ_c [Pa]	G_c [N/m]
1	Pyrolytic carbon	1500.0	22.50e9	2.50e9	45.0
2	TMPTA	1100.0	430.0e6	11.0e6	2.0
3	Alumina	3690.0	260.0e9	400.0e6	100.0

That said, the simulations will only use pyrolytic carbon due to time constraints. Pyrolytic carbon was also the material of choice of [64].

As for the possible choices of unit cell topologies, there are two: (1) the bitruncated octahedron (Figure 2.11) and (2) the octet (Figure 2.9).

The specimen geometry is summarised in Table 5.2.

Table 5.2: The geometric parameters for the truss-based lattices used in the simulations.

Parameter	Symbol [unit]	Value
Unit cell size	L [m]	5.0e-3
Specimen size	S [m]	50.0e-3
Specimen thickness	T [m]	25.0e-3
Relative density	$\bar{\rho}$ [-]	0.0125
Nº elements per strut	n_{strut} [-]	2
Fine region extension factor	f_{fine} [-]	0.50
Fine region subdivisions	N_f [-]	3

The unit cell size $L = 5.0 \cdot 10^{-3}$ m means each unit cell occupies a 5.0e-3 x 5.0e-3 x 5.0e-3 m space. The specimen size S and thickness T together mean the specimen has dimensions 50.0e-3 x 50.0-3 x 25.0e-3 m, and it comprises 500 unit cells. The relative density $\bar{\rho}$ is used to calculate the strut radius for the lattice according to Equation 5.1 (octet) and Equation 5.2 (bitruncated octahedron) depending on the unit cell choice. The fine region extension factor describes a square area around the impact zone where the mesh is locally refined. This fine region has a width equal to half the specimen size. With 3 fine region subdivisions, the number of elements per strut in the fine region is 8, while outside of the fine region, it is 2.

$$\begin{aligned} l_{\text{octet}} &= \frac{L}{\sqrt{2}} \\ r_{\text{octet}} &= \frac{1}{2}\sqrt{\frac{\bar{\rho}\sqrt{2}}{3\pi}} \cdot l_{\text{octet}} \end{aligned} \tag{5.1}$$

$$\begin{aligned} l_{\text{octahedron}} &= \frac{L}{2\sqrt{2}} \\ r_{\text{octahedron}} &= \frac{1}{2}\sqrt{\frac{\bar{\rho} \cdot 8\sqrt{2}}{3\pi}} \cdot l_{\text{octahedron}} \end{aligned} \tag{5.2}$$

In Equation 5.1 and Equation 5.2, l_{octet} and $l_{\text{octahedron}}$ are the strut lengths of the unit cell. r_{octet} is the strut radius of the octet and $r_{\text{octahedron}}$ is the strut radius of the bitruncated octahedron.

This now leaves the general simulation parameters, an overview of which has been presented in Table 5.3.

Table 5.3: General simulation parameters for the lattice impact simulation.

Parameter	Symbol [unit]	Value
Impactor radius	r_{impactor} [m]	5.0e-3
Impactor mass density	ρ_{impactor} [kg/m^3]	7850
Impactor velocity in z-direction	v_0 [m/s]	variable
Rigid contact penalty stiffness	γ [N/m]	1000
DG interface penalty	β_p and β_t [-]	100
Mode-mixity parameter	α [-]	1
Inter-beam contact penalty stiffness factor	R_{contact} [-]	variable
Time step	Δt [s]	1.0e-10
Final time	t_{final} [s]	variable

In each simulation, a steel sphere collides with the specimen in the centre of the specimen. As seen from Table 5.3, the radius of the impactor is 5.0 mm, which is equal to the size of a unit cell. For simplicity, this impactor is modelled as an analytical rigid body, allowing it to be represented in the system as a special boundary condition rather than needing to be discretised.

Of the parameters in Table 5.3, v_0 is a variable which is chosen for each simulation run, and accordingly, t_{final} is also chosen for each simulation run. Table 5.3 is meant to provide an overview of the general simulation parameters, and aside from v_0, t_{final}, R_{contact} and possibly γ, these inputs remain the same between simulation runs and test cases. γ can change depending on the unit cell topology since the difference in stiffness between the two unit cell types can affect the rigid contact behaviour. Similarly, R_{contact} may need to be calibrated for the initial impactor velocity since higher impactor velocities can cause higher velocity fragments whose contact events with other members in the lattice may be missed if the penalty is too low. The contact penalty stiffness is calculated using Equation 4.4.

The specific simulation inputs for the different cases are given below in Table 5.4. The intention was to simulate each test case from Table 5.4 is for an impactor velocity v_0 ranging from 50 m/s to 350 m/s. However, due to time constraints, the bitruncated octahedron test cases were only run with v_0 = 150 m/s and v_0 = 350 m/s, while the octet test cases were only run with v_0 = 350 m/s.

Table 5.4: The specific input parameters used by each test case.

Input	Case 1A	Case 1B	Case 2A	Case 2B	Case 3A	Case 3B
Initial impactor velocity	350	350	350	350	150	150
Cell topology	1	1	2	2	1	1
R_{contact} [-]	1000.0	-	1000.0	-	1000.0	-
t_{final} [s]	0.15e-3	0.15e-3	0.15-e3	0.15e-3	0.25e-3	0.25e-3

All test cases denoted by a B in their designation are essentially the same as the cases designated by an A, except these simulations do not model beam-to-beam contact. This is why these cases have "-" for R_{contact} because they do not have self-contact. This way, they act as the control group for comparing results with and without beam contact. Cell topology 1 refers to the bitruncated octahedron and cell topology 2 refers to the octet. One can also see why t_{final} is a variable, because the higher impact velocities do not require as long of a simulation. Additionally, γ for all test cases was 1000 and $\bar{\rho}$ was 0.0125. Finally, as a reminder, the material for each test case was pyrolytic carbon (see Table 5.1).

The initial impactor velocity is applied to the rigid impactor and the velocity vector points in the positive z-direction. As for the boundary conditions, the faces on the external boundary of each specimen in the x-y plane have been fixed. This is to say, the nodes on the planes $x = 0$, $x = S$, $y = 0$ and $y = S$ have their positions fixed in all three dimensions.

Figure 5.1 shows the initial state of case 1A.

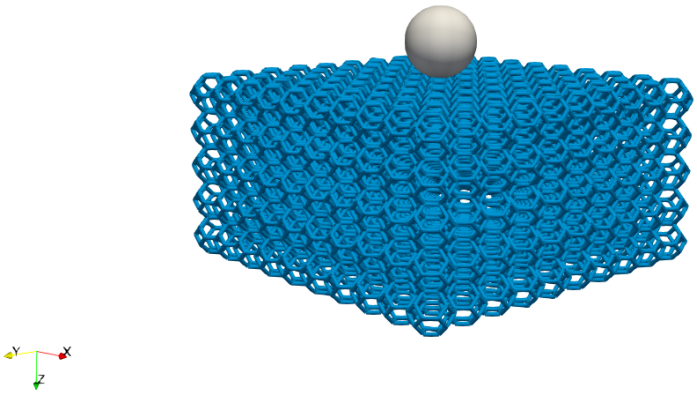


Figure 5.1: The set-up for test case 1A. The impactor is propelled in the positive z-direction. The thickness of the struts has been exaggerated for visual clarity.

Note that the impactor moves in the positive z-direction, meaning it comes from the negative z-direction. Furthermore, the centre of the impactor is positioned at $\boldsymbol{x} = (0.025, 0.025, -0.005)$ m at $t = 0$ s. From Figure 5.1, the top plane of the lattice is located at a height of $z = 0$ m. The initial states of all other test cases are similar to case 1A: the impactor is located at the same initial position and it moves in the positive z-direction.

5.2. Collision of a rigid impactor with a truss lattice

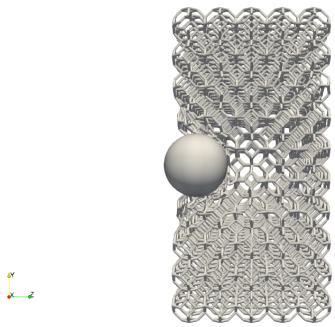
The results of the simulation cases listed in Table 5.4 are presented in this section. First, visualisations of each test cases are illustrated using some frames captured in Paraview. Then, the impactor trajectory results for each test case are shown. This is done through the time histories of the impactor z-positions and z-velocities for different initial velocities. After the trajectory results, the energy response of the truss lattice and impactor system is presented for different initial velocities and for the cases where inter-beam contact is and is not modelled. The aim is to evaluate how beam-to-beam contact influences the energy response.

5.2.1. Impact visualisations

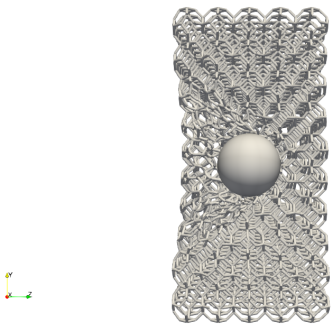
Visualisations of the impact simulations for each of the test cases listed in Table 5.4 are shown below. These have been generated using Paraview and the VTK files written during each simulation. The VTK files are written with a frequency of 400 kHz. To help with visibility, the lattices were sectioned along their mid-plane so that one can see the impactor move through them. Note that for all simulations, the simulations were stopped the moment the impactor's z-position satisfied the condition $z > T + r_{\text{impactor}}$ where T is the lattice thickness (in the z-direction) and r_{impactor} is the impactor's radius.

Case 1A

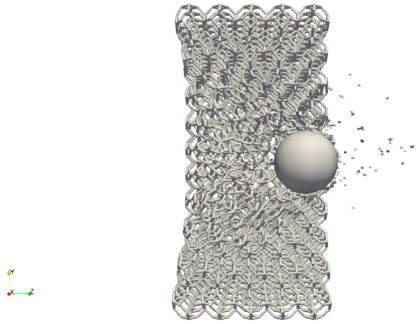
Case 1A uses bitruncated octahedron unit cells and inter-strut contact is modelled. The impactor has an initial velocity of 350.0 m/s. The deformed state of the lattice at select points in time have been included as Figure 5.2.



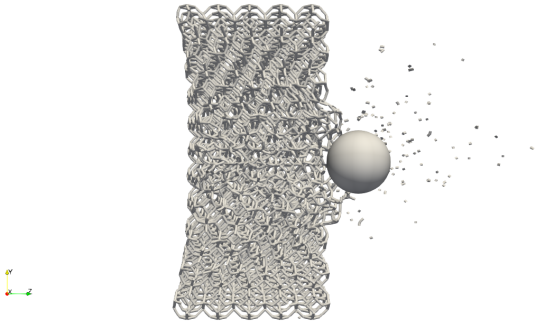
(a) Simulation frame for case 1A captured at approximately 25 μs.



(b) Simulation frame for case 1A captured at approximately 50 μs.



(c) Simulation frame for case 1A captured at approximately 75 μs.



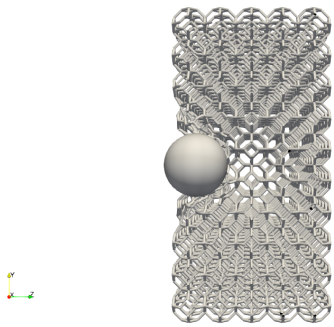
(d) Simulation frame for case 1A captured at approximately 100 μs.

Figure 5.2: Four simulation frames for case 1A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

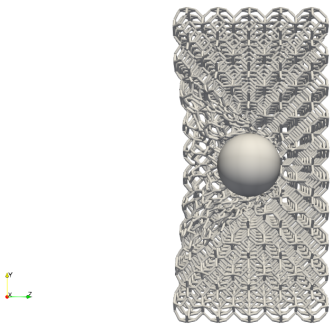
From Figure 5.2 one can observe complete penetration of the lattice by the impactor. One can see from Figure 5.2b to Figure 5.2d how the impactor causes large local deformations in a zone behind it. This leads to fracture of the lattice members and eventually to ejection of the fragmented struts.

Case 1B

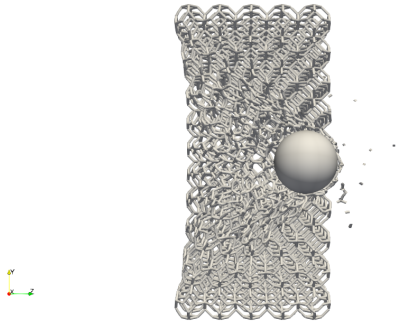
Case 1B also uses bitruncated octahedron unit cells but inter-strut contact is not modelled. The impactor has the same initial velocity of 350.0 m/s. The deformed state of the lattice at select points in time have been included as Figure 5.3.



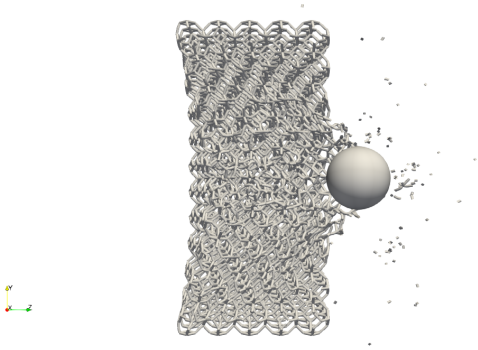
(a) Simulation frame for case 1B captured at approximately 25 μs.



(b) Simulation frame for case 1B captured at approximately 50 μs.



(c) Simulation frame for case 1B captured at approximately 75 μs.



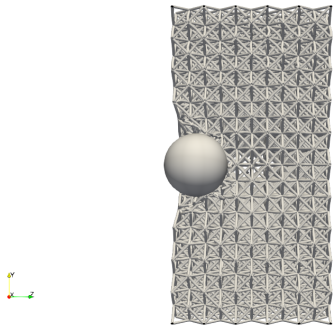
(d) Simulation frame for case 1B captured at approximately 100 μs.

Figure 5.3: Four simulation frames for case 1B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

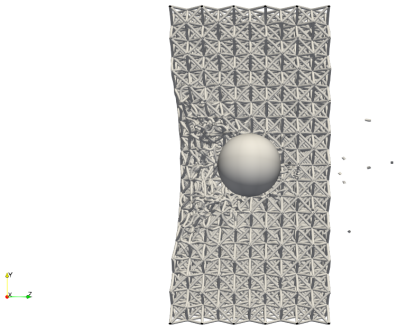
The frames in Figure 5.3 tell a similar story as Figure 5.2. The impactor perforates the lattice, and one can see the same progression of large localised deformations behind the impactor, followed by strut failure and fragmentation. Visually speaking, there appears to be little difference in impact response between case 1B, without self-contact, and case 1A, with self-contact. Although there may be more fragments visible in Figure 5.2d. This is actually a common theme with the other simulation runs: the eye test reveals few differences, but this is also why the trajectory and energy plots were generated. These will be covered in Subsection 5.2.2 and Subsection 5.2.3.

Case 2A

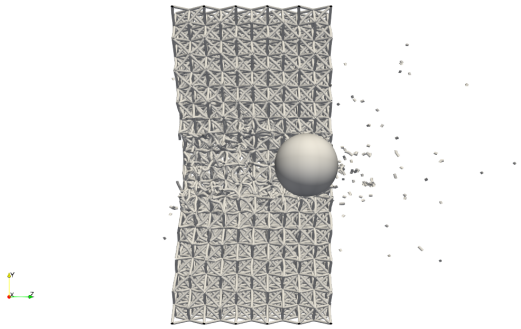
Case 2A comprises octet unit cells and self-contact is modelled in this case. The impactor has an initial velocity of 350.0 m/s. The deformed state of the lattice at select points in time are shown below as Figure 5.4.



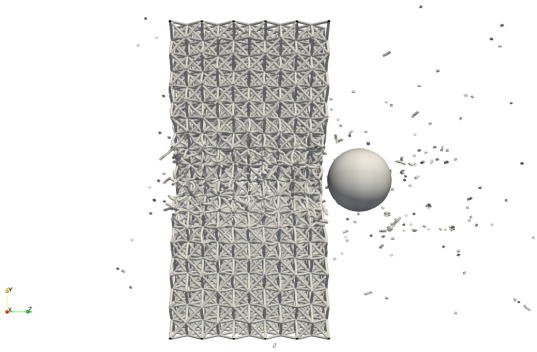
(a) Simulation frame for case 2A captured at approximately 25 μs.



(b) Simulation frame for case 2A captured at approximately 50 μs.



(c) Simulation frame for case 2A captured at approximately 75 μs.



(d) Simulation frame for case 2A captured at approximately 100 μs.

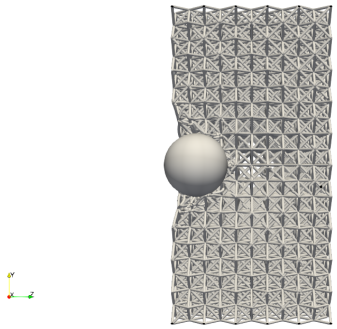
Figure 5.4: Four simulation frames for case 2A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

Figure 5.4 shows complete penetration. In fact, in all cases the lattice was perforated by the impactor. Compared to case 1A and case 1B, the deformations for the octet lattice seem to be not as large. This is especially apparent from Figure 5.4d. The deformations in the last frame for the octet appear to be less severe and more localised around the impactor's path compared to Figure 5.2d and Figure 5.3d where one can see that a large part of the lattice has deformed. This is likely the result of the octet unit cell being stretch-dominated while the bitruncated octahedron unit cell is bending-dominated [6].

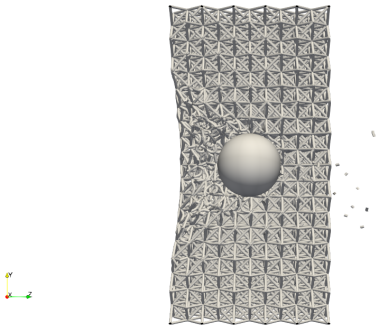
Because the octet is stretch-dominated, a larger fraction of the absorbed kinetic energy is turned into fracture energy and a lower fraction of the absorbed kinetic energy goes into elastic deformations compared to the bitruncated octahedron [100]. This means the balance between the contributions from fracture and elastic deformations to the absorption of the impactor's kinetic energy is more in favour of fracture compared to the bitruncated octahedron. This is also similar to what Butruille et al. (2024) found when comparing the energy dissipation mechanisms of octet lattices with truncated octahedron lattices [12]. In practical terms, the ratio of W_{frac} to W_{int} for the octet lattices will be higher than for the bitruncated octahedron lattices. This can also be observed from Figure 5.14 and Figure 5.16.

Case 2B

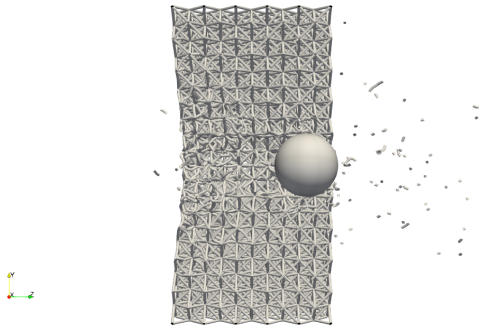
Case 2B is the no beam-contact counterpart to case 2A. The deformed state of the lattice at select points in time are shown below as Figure 5.5.



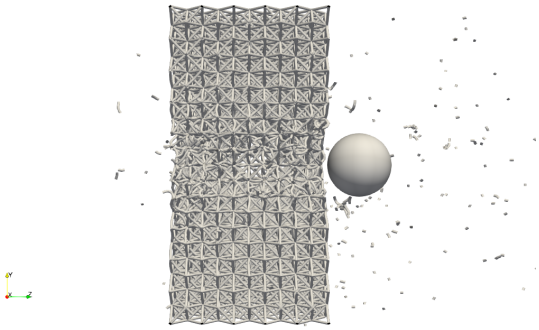
(a) Simulation frame for case 2B captured at approximately 25 μs.



(b) Simulation frame for case 2B captured at approximately 50 μs.



(c) Simulation frame for case 2B captured at approximately 75 μs.



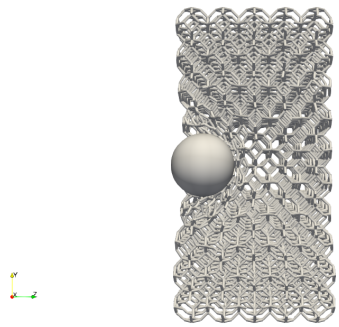
(d) Simulation frame for case 2B captured at approximately 100 μs.

Figure 5.5: Four simulation frames for case 2B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

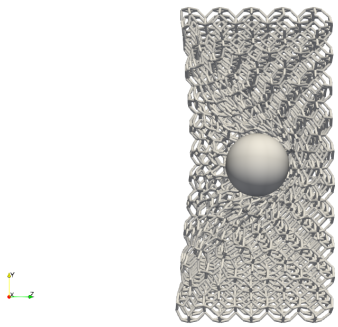
Just as with case 1B, running the simulation without accounting for self-contact largely results in a similar deformation response. When Figure 5.5 is compared with Figure 5.4 one can see that the deformations in the no beam-contact case also remain localised to the impact zone and these deformations are not as large as seen in the case of the bitruncated octahedron lattices.

Case 3A

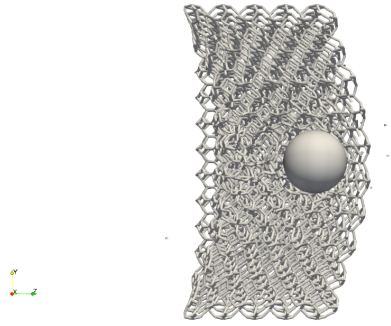
Case 3A consists of bitruncated octahedron unit cells and it accounts for contact between lattice members. The impactor has an initial velocity of 150.0 m/s this time around. The deformed configuration of the lattice at select points in time can be seen in Figure 5.6.



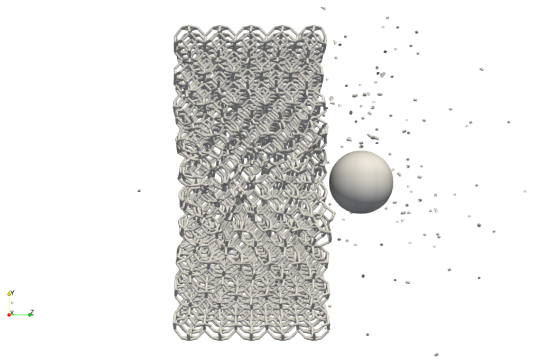
(a) Simulation frame for case 3A captured at approximately 62.5 μs.



(b) Simulation frame for case 3A captured at approximately 125 μs.



(c) Simulation frame for case 3A captured at approximately 187.5 μs.



(d) Simulation frame for case 3A captured at approximately 245 μs.

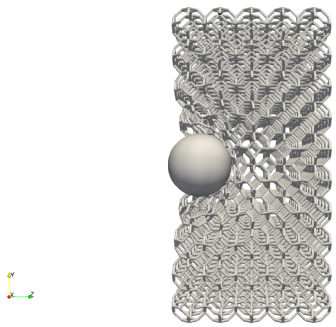
Figure 5.6: Four simulation frames for case 3A. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

Even if $v_0 = 150.0$ m/s, Figure 5.6 still shows that the lattice is perforated. However, there are differences with Figure 5.2. What is still the same is that one can observe a similar process of large deformations in the wake of the impactor, resulting in fracture of the truss members and subsequent ejection of the fragments. The difference lies in that the impactor causes deformations in front of it. This is probably because the deformations can propagate faster through the lattice than the impactor now that the velocity of the impactor is 150.0 m/s initially rather than 350.0 m/s. As a result, unit cells downstream of the impactor can be deformed because the unit cells closer to the impactor deform and are pushed out by the impactor, pushing the unit cells further downstream away too. When the impactor travels much faster, this propagation of the deformations downstream does not seem to be able to happen before the impactor hits the unit cells downstream.

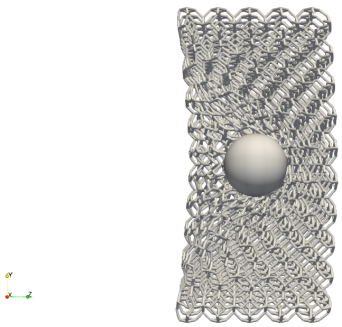
Something else which stands out from Figure 5.6 is that in Figure 5.6c, one can see that the deformations reach the fixed boundary condition on the zy-plane as evidenced by the column of unit cells which do not seem to have deformed. This means that the lattice is slightly too small to prevent boundary effects. If similar large-scale truss-metamaterial impact simulations are run in future work, then one recommendation is to use a larger lattice than was used in this thesis to ensure no boundary effects.

Case 3B

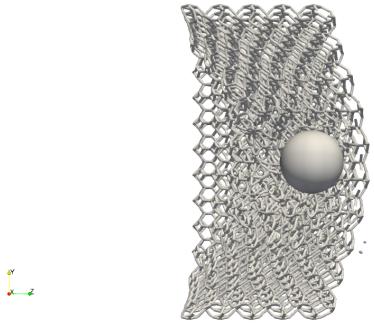
Case 3B is the no beam-contact version of case 3A. The deformed configuration of the lattice at select points in time can be seen in Figure 5.7.



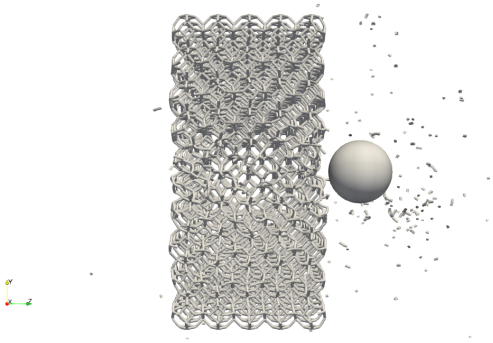
(a) Simulation frame for case 3B captured at approximately 62.5 μs.



(b) Simulation frame for case 3B captured at approximately 125 μs.



(c) Simulation frame for case 3B captured at approximately 187.5 μs.



(d) Simulation frame for case 3B captured at approximately 247.5 μs.

Figure 5.7: Four simulation frames for case 3B. Each frame corresponds to approximately 2.5 μs. The lattice was sectioned along the mid-plane.

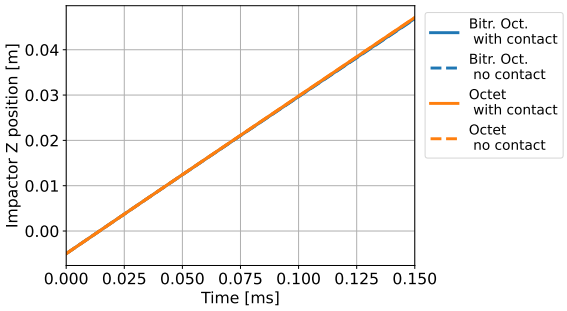
Just as with the previous test cases, when the simulation is run without accounting for contact between the struts, then the deformations do not look all that different. Figure 5.7 still shows the sequence of large local deformations behind the impactor, leading to strut failure and eventual fragmentation. One can also see that Figure 5.7c is similar to Figure 5.6c in that the impactor causes deformations downstream too.

One can also observe from Figure 5.7c that there are probably some boundary effects as the same column of undeformed unit cells can be seen, similar to Figure 5.6c.

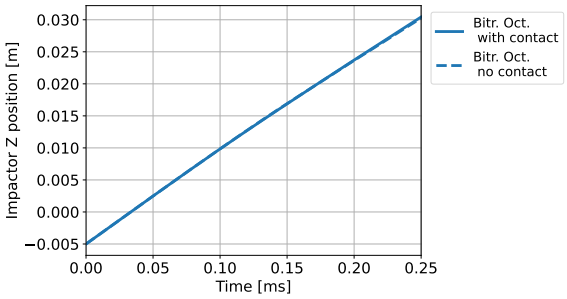
5.2.2. Trajectory plots

After presenting the simulation frames, the trajectory plots are treated now. These are the time histories of the impactor's z-position and z-velocity. Each trajectory plot is an overlay of all lattice types: bi-truncated octahedron or octet, and whether or not beam contact is modelled for a given initial velocity. However, for v_0 = 150.0 m/s, there is no octet trajectory data because there was not enough time to simulate test cases for v_0 = 150.0 m/s with the octet lattice.

Figure 5.8 illustrates the time histories of the impactor z-position for the different test cases.



(a) The time history of the z-position of the impactor for an initial velocity $v_0 = 350.0$ m/s.



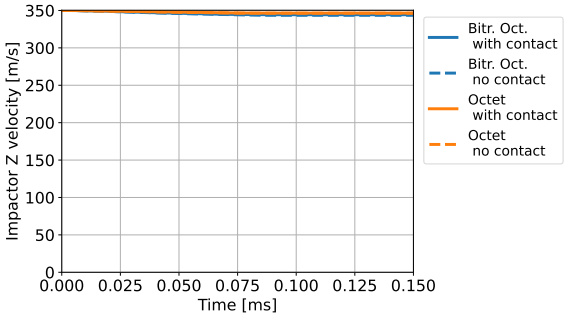
(b) The time history of the z-position of the impactor for an initial velocity $v_0 = 150.0$ m/s.

Figure 5.8: The time history of the z-position of the impactor at initial velocity $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s for all cases. Note that the impactor starts off at a z-position of -0.005 m.

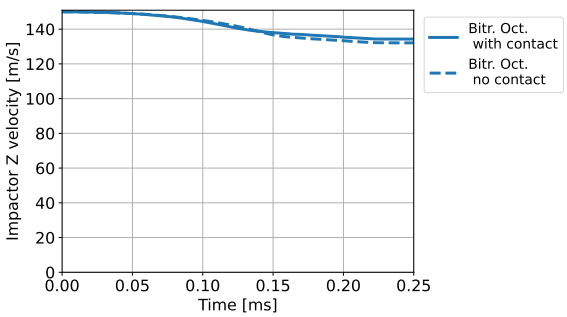
One can see in Figure 5.8a and Figure 5.8b that for both initial velocities, the impactor perforates the lattice since the z-position of the impactor for every case exceeds $z = 0.025$ m, the lattice height, at the last time step. That said, Figure 5.8b only shows the bitruncated octahedron lattice since there was no time to run simulations at $v_0 = 150.0$ m/s for the octet lattice. As such, one cannot be certain whether the octet lattice would have been perforated like the bitruncated octahedron lattice when $v_0 = 150.0$ m/s.

In any case, the z-position plots do not show noticeable differences in the position response between the cases which do model beam contact and the cases which do not. So, the presence or absence of modelling contact between the struts was not enough at these initial velocities to change the qualitative impact response from perforation to, for instance, capture or rebound. However, there are observable differences in the velocity response when contact between the beams is modelled or not.

Figure 5.9 shows the time histories of the impactor's z-velocity for the different test cases. To make the results better legible and distinguishable, the same plots were generated without restricting the y-axis and included as Figure 5.10.

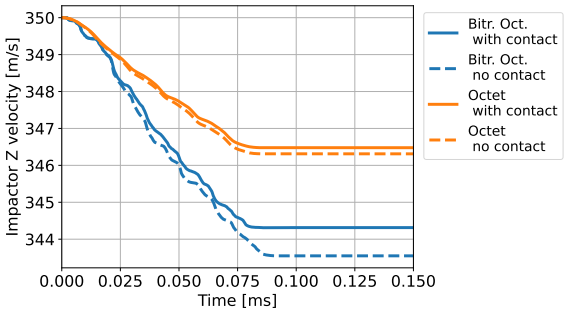


(a) The time history of the z-velocity of the impactor for an initial velocity $v_0 = 350.0$ m/s.

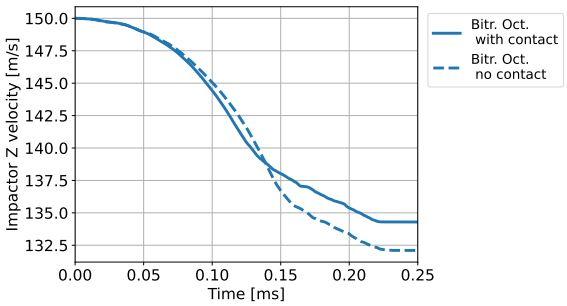


(b) The time history of the z-velocity of the impactor for an initial velocity $v_0 = 150.0$ m/s.

Figure 5.9: The time history of the z-velocity of the impactor at initial velocity $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s for all cases.



(a) A more legible version of the time history of the z-velocity of the impactor for an initial velocity $v_0 = 350.0$ m/s.



(b) A more legible version of the time history of the z-velocity of the impactor for an initial velocity $v_0 = 150.0$ m/s.

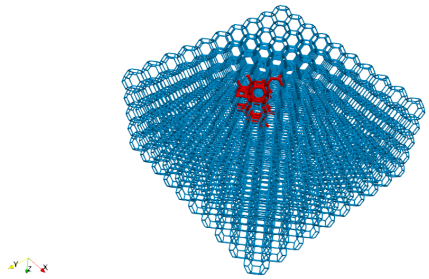
Figure 5.10: Better legible versions of the time history of the z-velocity of the impactor at initial velocity $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s for all cases.

Figure 5.10a indicates that capturing contact between the struts reduces the capacity of the lattice to slow down the impactor, both for the bitruncated octahedron and the octet topologies. Figure 5.10b implies the same for the bitruncated octahedron, although a lack of data for the octet at this velocity makes it impossible to say for certain whether the octet lattice behaves the same.

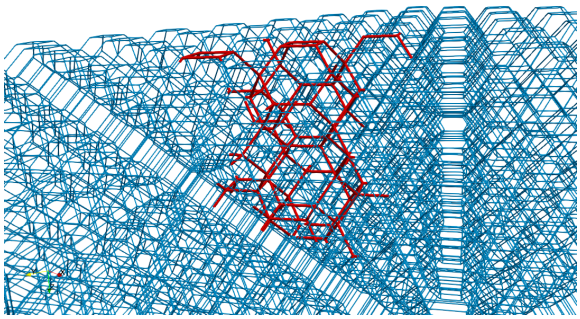
Because the impactor maintains a higher final velocity when inter-beam contact is accounted for, this means it has a higher final kinetic energy. If the final kinetic energy of the impactor is higher than the no beam-contact cases, then that must mean the absorbed energy, given by the difference in final and initial impactor kinetic energy (see Subsection 2.2.2), is lower for the beam-contact cases. This is contrary to the expectation that modelling self-contact of the lattice improves the energy absorption performance since it allows for capturing an additional impact mechanism otherwise not accounted for when contact between the struts is not modelled. This additional impact mechanism is compaction.

As mentioned in Subsection 2.3.4 and demonstrated in D and E of Figure 2.20, compaction is a stiffening effect induced by contact between neighbouring lattice members. Therefore, self-contact enables the lattice to exhibit compaction behaviour, which gradually increases its stiffness as it is compressed. Furthermore, self-contact also allows for distributing the kinetic energy imparted during impact over a larger region because struts which contact each other can transfer kinetic energy to each other. As such, one would expect that modelling self-contact of the lattice would result in improved impact performance.

Yet, Figure 5.10 suggest the opposite: capturing contact between lattice members reduces the energy absorption performance of the lattice compared to when no inter-beam contact is modelled. There are a couple of possible reasons why the impact response with self-contact is not as expected. The first possible cause is that compaction could not be captured to the same extent as in Figure 2.20 because the size of the partitions are too small. Figure 5.11 provides visuals for one such partition for case 1A. Keep in mind that the other test cases have similar partitions.



(a) The size of a partition (red) in the impact region compared to the full lattice size of case 1A. The thickness of the struts in the partition has been exaggerated for visual clarity.



(b) A close-up of the partition from Figure 5.11a. One can see it consists of just a couple of unit cells.

Figure 5.11: The size of a single partition for case 1A. The partition is coloured in red and the thickness of the truss members has been emphasised.

Figure 5.11a illustrates the size of a partition relative to the rest of the lattice and Figure 5.11b offers a close-up view of the same partition. One can see in Figure 5.11b that a single partition comprises a couple of unit cells. The fact that the partitions are relatively small compared to the rest of the truss metamaterial is an important clue when one recalls from Subsection 3.2.4 that the contact model can only capture beam-to-beam contact within the same partition. This is the result of a limitation of the implementation of the contact model to the effect that different CPUs do not have knowledge of elements from other partitions that could form contact pairs with elements in their own partition. This means the current implementation of the contact model only support intra-partition contact. Therefore, struts of different partitions that in real life would be in contact with each other, would not be considered to be in contact with each other in this computational framework, essentially resulting in missed contact events.

Because inter-partition contact cannot be modelled by the beam-contact-fracture framework, the ability to capture densification is contingent on the partition size. If the partitions are large, then capturing compaction is still possible because more contact events can then occur within the same partition leading to fewer missed contact events caused by inter-partition contact. For instance, if the whole impact zone around the impactor were a single partition, then most contact events caused by the impact would happen within this partition, meaning fewer missed contact events due to inter-partition contact.

As such, one way to capture compaction with the current implementation of the beam-contact-fracture framework would be to use larger partitions. However, increasing the partition size, so decreasing the number of partitions, would slow down the simulation too. A single simulation takes upwards of a day or more (depending on Δt and v_0), so using larger partitions and thus fewer cores could exponentially increase the computational time. Instead, the best course of action would be to fully parallelise the contact model. Hence why it is a strong recommendation for future work.

However, even if densification cannot be fully modelled, that does not explain why the absorption performance degrades because of self-contact. Being unable to capture densification ought to mean simulations that capture self-contact do not display better absorption performance, but the fact that they show worse performance could mean self-contact causes impact phenomena that are detrimental to the impact response. One possible explanation is that the fragments produced by the collision between the impactor and the struts are propelled into intact struts, in other words: fragmentation-induced fracture. Their contact events then can cause fracture of those struts, weakening the structure downstream of the impactor and resulting in less energy absorption.

That said, this would require the fragments to move faster than the impactor itself. From a conservation of momentum point of view, it is possible. After all, the impactor has a much larger mass than the beam fragments, and the impactor does lose momentum since its velocity decreases. The ensuing impulse exchange is capable of sending beam fragments flying at a higher velocity. However, showing this in

visualisations would provide actual proof. This is why a sectioned view was generated in Paraview which shows the velocity with a colour map. For case 1A, one can find this sectioned velocity view below as Figure 5.12.

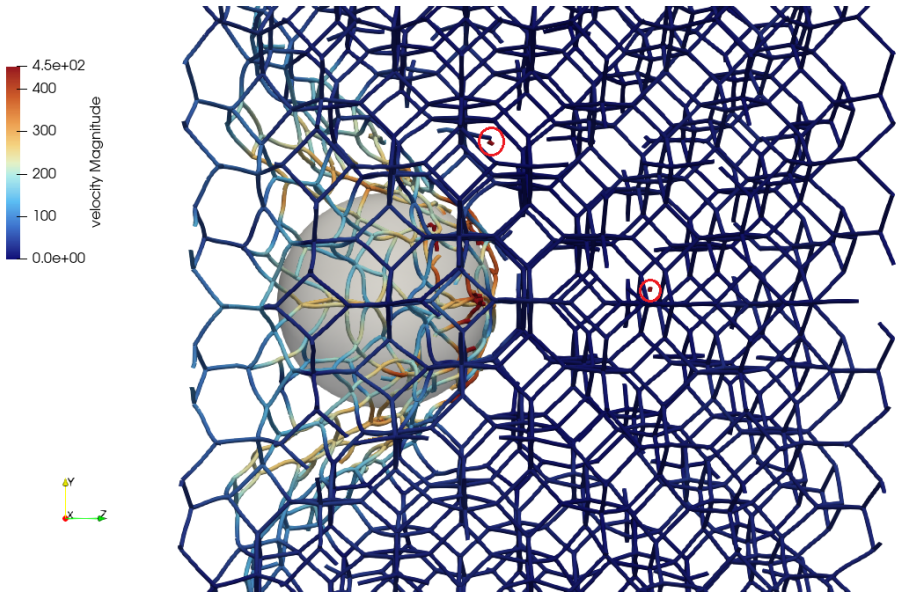


Figure 5.12: A sectioned view of the lattice from case 1A to show that fragments can reach velocities higher than the impactor's velocity. Highlighted by red bubbles are some fragments which have been propelled into the structure following impact. Note that this sectioned view only shows a thin strip of the lattice for visibility reasons.

In Figure 5.12, one can observe from the colour map that some members do reach velocities higher than the impactor's velocity. Furthermore, this specific frame even shows some fragments which have shot off into the surrounding structure. With this, one can conclude that it is possible that the impactor can cause fragments to be propelled downstream.

Still, even if one can prove that the impactor causes fragments to be shot into the surrounding structure, another point of contention with the hypothesis of fragmentation-induced fracture is that it requires the fragments to collide with the pristine struts downstream. The problem is that it is difficult to prove such contact events actually occur because the VTK files are only updated with a frequency of 400 kHz. With $\Delta t = 1 \cdot 10^{-10}$ s, this means the output files are updated every 25000 time steps, making it very difficult to capture contact events between fragments and struts in the visualisations. On top of this, one must also remember that inter-partition contact is not modelled, so this fragmentation-induced fracture can only occur within a single partition. Fragments scattered from one partition will not be able to make contact with pristine struts in other partitions, so if fragmentation-induced fracture does happen, then it occurs as the impactor moves through the partitions in its line of action (the z-axis).

5.2.3. Energy response

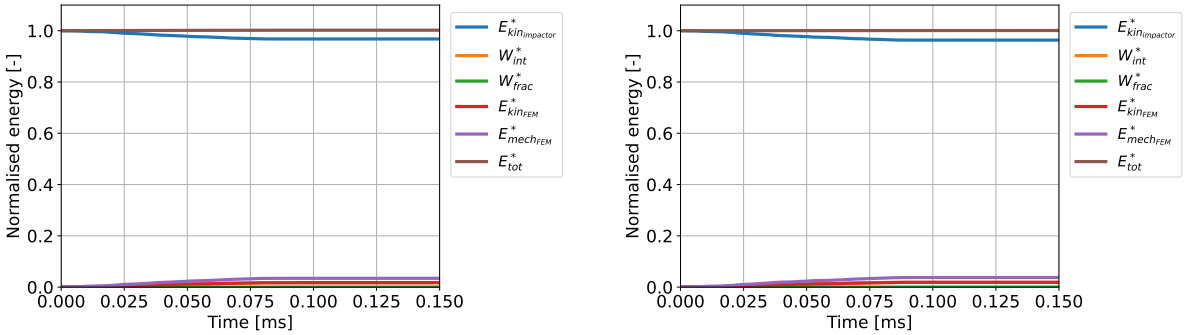
The last collection of results pertain to the energy response of the system for each test case. In accordance with the second part of subtask 3: evaluating the effect of self-contact on the energy absorption, the energy behaviour of each test case will be analysed in the following. To this end, the energy components and the absorbed energy were plotted for each test case.

Energy components

The energy components were plotted mainly to verify whether the system maintains energy balance. The plots are grouped by lattice type and initial impact velocity, so Figure 5.13 are for $v_0 = 350.0$ m/s and bitruncated octahedron lattices, Figure 5.15 are for $v_0 = 350.0$ m/s and octet lattices, and Figure 5.17 are for $v_0 = 150.0$ and bitruncated octahedron lattices. To clarify the legend, $E^*_{kin_{impactor}}$ is the kinetic energy of the impactor normalised by the initial kinetic energy of the impactor. The * signifies the

energy component is normalised by the initial impactor kinetic energy. Similarly, W^*_{int} is the normalised internal energy, W^*_{frac} is the normalised fracture energy, $E^*_{kin_{FEM}}$ is the normalised kinetic energy of the lattice, $E^*_{mech_{FEM}}$ is the normalised mechanical energy of the lattice and E^*_{tot} is the normalised energy of the system (impactor and lattice).

Figure 5.13 shows the time history of the energy components of case 1A and case 1B.

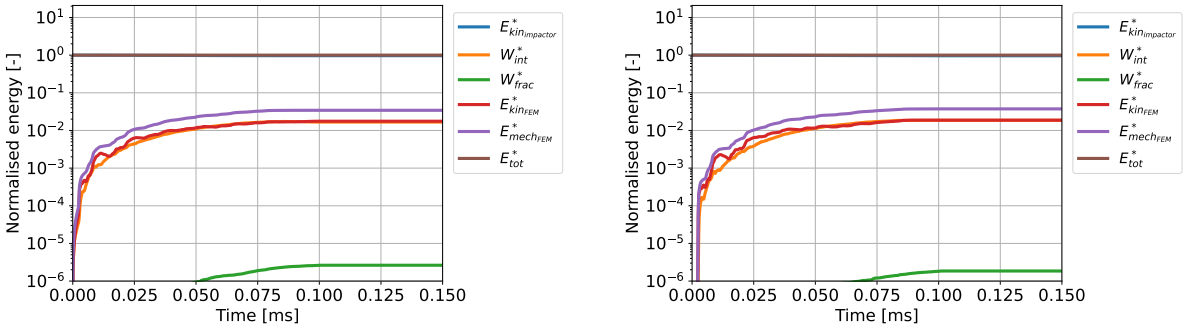


(a) The time history of the energy components of the system for case 1A at an initial velocity $v_0 = 350.0$ m/s.

(b) The time history of the energy components of the system for case 1B at an initial velocity $v_0 = 350.0$ m/s.

Figure 5.13: The time history of the energy components of the system at initial velocity $v_0 = 350.0$ m/s for cases 1A and 1B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy.

One can see from Figure 5.13a and Figure 5.13b that the total energy of the system for case 1A and case 1B are balanced, as evidenced by the fact that E^*_{tot} is constant for both cases. However, because of the high impactor velocity, the impactor kinetic energy is much larger than the energy components of the lattice, skewing the plot. In order to make the smaller energy components better distinguishable, Figure 5.14 was generated, which features a logarithmic y-axis. Note that a lower limit of 10^{-6} was used for the y-axis because the energy components of the lattice are initially 0.0, which would skew the semi-log plot and make the plot as a whole difficult to read.



(a) The time history of the energy components of the system for case 1A at an initial velocity $v_0 = 350.0$ m/s with logarithmic scaling of the y-axis.

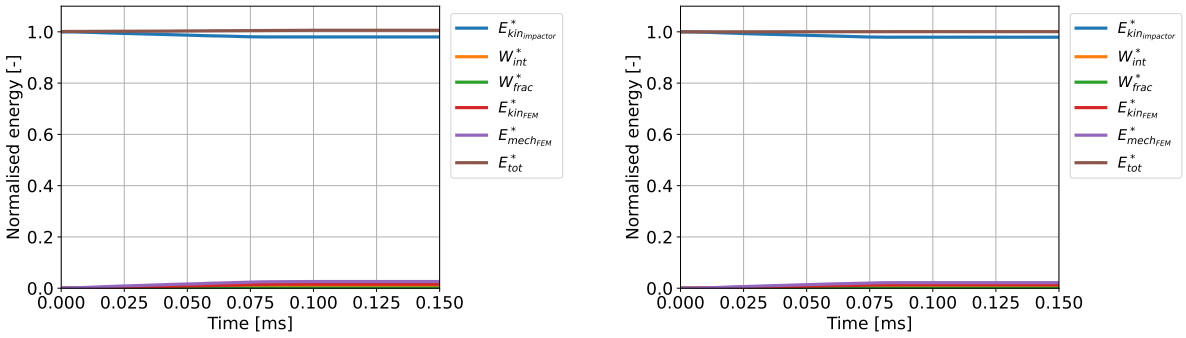
(b) The time history of the energy components of the system for case 1B at an initial velocity $v_0 = 350.0$ m/s with logarithmic scaling of the y-axis.

Figure 5.14: The time history of the energy components of the system at initial velocity $v_0 = 350.0$ m/s for cases 1A and 1B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable.

Now that one can distinguish the energy components of the lattice more easily, it becomes easier to compare the energy response between case 1A, with beam contact, and case 1B, without beam contact. What stands out the most is that for case 1A, the normalised fracture energy rises past 10^{-6} earlier for $t > 0.050$ ms, while for case 1B, it takes until $t \approx 0.065$ ms. The normalised fracture energy for case 1A is also higher than for case 1B. This can be an indication that the fragmentation-induced fracture phenomenon supposed in Subsection 5.2.2 did occur because the case with self-contact has more fracture energy, meaning more fragments, and these fractures occur earlier than the case without

self-contact. The fact that fractures occur earlier with self-contact could mean that fragments propelled by the impactor into the pristine struts downstream caused fracture in said struts. When combined with the fact that the fracture energy is higher for the case with self-contact, it lends credence to hypothesis that the fragmentation-induced fracture weakened the structure downstream of the impactor, leading to reduced energy absorption performance.

The energy time histories for case 2A and case 2B are plotted in Figure 5.15. Figure 5.16 is the semi-log version to make the lattice energy components more easily distinguishable.

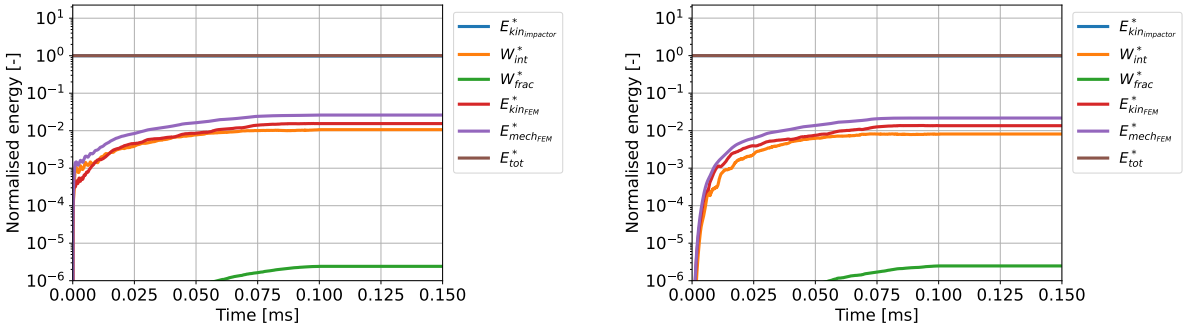


(a) The time history of the energy components of the system for case 2A at an initial velocity $v_0 = 350.0$ m/s.

(b) The time history of the energy components of the system for case 2B at an initial velocity $v_0 = 350.0$ m/s.

Figure 5.15: The time history of the energy components of the system at initial velocity $v_0 = 350.0$ m/s for cases 2A and 2B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy.

Figure 5.15 demonstrates that the total energy for case 2A and case 2B is balanced. Other than this, Figure 5.15 do not tell much, which is why Figure 5.16 was included.



(a) The time history of the energy components of the system for case 2A at an initial velocity $v_0 = 350.0$ m/s with logarithmic scaling of the y-axis.

(b) The time history of the energy components of the system for case 2B at an initial velocity $v_0 = 350.0$ m/s with logarithmic scaling of the y-axis.

Figure 5.16: The time history of the energy components of the system at initial velocity $v_0 = 350.0$ m/s for cases 2A and 2B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable.

Figure 5.16 are the energy time histories of case 2A and case 2B when the y-axis has logarithmic scaling. This way, one can actually see the different energy components of the lattice. Unlike with case 1A and case 1B, the fracture energy with and without self-contact are roughly the same. Similarly, one can see from Figure 5.16a and Figure 5.16b that the normalised fracture energy rises above 10^{-6} at about the same time. This seems to indicate that for the cases with octet lattices, fracture-induced fragmentation may not have been that prevalent. This also explains why the loss in absorption performance from the octet lattice was comparatively less than the loss in absorption performance from the bitruncated octahedron lattice seen in Figure 5.10a.

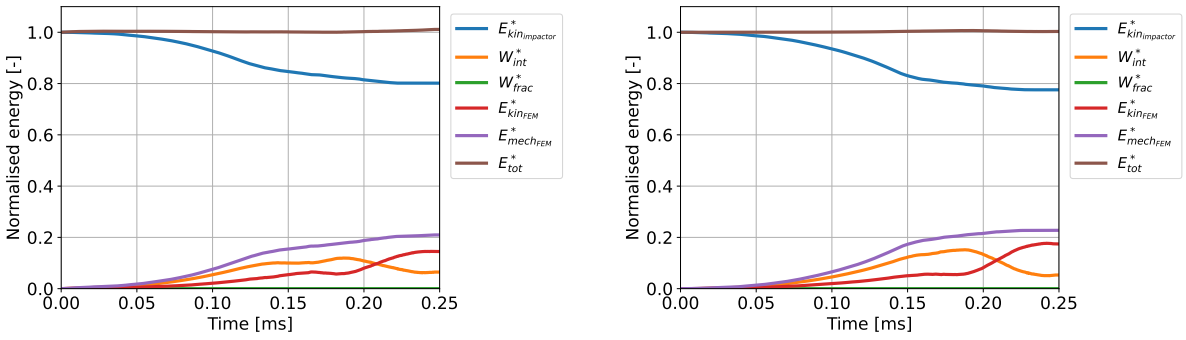
Furthermore, when the ratio of W^*_{frac} to W^*_{int} is compared between case 2A and case 1A, and between case 2B and case 1B, one will see that said ratio is larger for the octet lattices. This lines up with what was discussed in Subsection 5.2.1 as to why the octet lattices exhibit less severe deformation than the bitruncated octahedron lattices.

Lastly, one will notice a difference in the energy time history of the lattice when comparing Figure 5.16a to Figure 5.16b. It appears that the kinetic and internal energy of the lattice rise much earlier when inter-strut contact is modelled compared to when it is not. This could be an indication that the octet lattice with self-contact does manage to capture compaction, at least within a partition. As discussed in Subsection 5.2.2, larger partitions allow for more possible contact events to be captured within a partition. However, if a unit cell has a denser network of struts, then that could also lead to more possible contact events. Unlike the bitruncated octahedron (see Figure 2.11), which has no internal struts, the octet (see Figure 2.9) does. The internal struts of octet cells could allow for more possible contact events within a single partition, which may even make compaction behaviour possible for the octet lattice.

However, this compaction behaviour at the start of the impact simulation does not seem to have had much effect on the absorption of the kinetic energy of the impactor towards the end of the simulation as evidenced by the fact that the energy absorption performance of case 2A never exceeds case 2B (see Figure 5.10a). This may be explained by the fact that inter-partition contact is not captured by the contact model, so the lattice as a whole does not exhibit densification, but only locally.

Additionally, as mentioned in Subsection 2.2.2, Butruille et al. (2024) explain that for octet lattices, fracture is a relatively larger contribution to the absorbed energy than compaction, unlike for bitruncated octahedron lattices [12]. This could also explain why the loss of energy absorption for case 2A compared with case 2B was not as high as that for case 1A compared with case 1B. The inability of the computational framework to properly capture compaction of the entire lattice has affected the octet lattice less than the bitruncated octahedron lattice.

The last set of energy time history plots belong to case 3A and case 3B. Figure 5.17 shows the energy components with regular axis scales while Figure 5.18 uses logarithmic scaling on the y-axis.



(a) The time history of the energy components of the system for case 3A at an initial velocity v_0 = 150.0 m/s.

(b) The time history of the energy components of the system for case 3B at an initial velocity v_0 = 150.0 m/s.

Figure 5.17: The time history of the energy components of the system at initial velocity v_0 = 150.0 m/s for cases 3A and 3B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy.

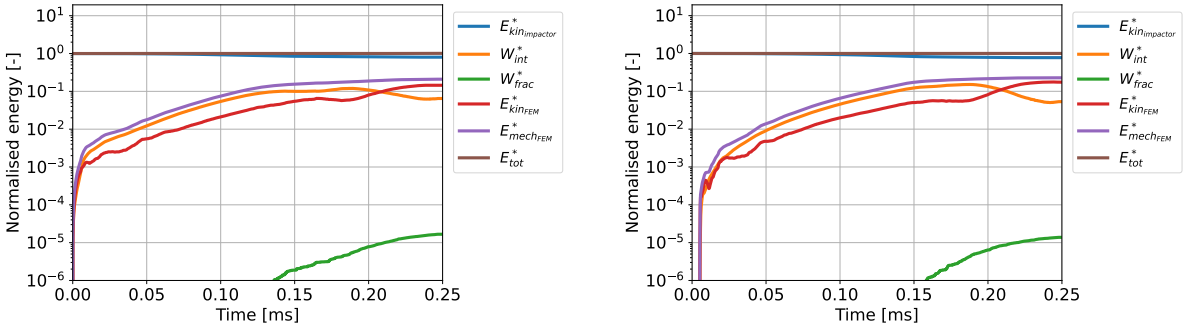
As with the other test cases, Figure 5.17 shows that the total energy of the system stays more or less balanced although there are some barely visible fluctuations. Unlike with Figure 5.13 and Figure 5.15, the impactor kinetic energy is low enough to not skew the plot entirely, making some of the energy components of the lattice distinguishable, with the exception of the fracture energy. Figure 5.17 tells a similar story as Figure 5.10b: towards the end of the simulation the no beam-contact case shows better energy absorption performance than the beam-contact case. This is apparent from the fact that $E^*_{mech_{FEM}}$ near the end for case 3B is higher than for case 3A. One can also see that $E^*_{kin_{impactor}}$ is lower for case 3B

near the end compared to case 3A.

Another consequence of the impactor kinetic energy skewing the plot less is that compared with Figure 5.13 and Figure 5.15, the energy conversion from internal to kinetic energy in the lattice can be observed without logarithmic scaling. This can be seen in Figure 5.17a and Figure 5.17b for $t > 0.20$ ms, where the W^*_{int} and the $E^*_{kin_{FEM}}$ curves cross each other. This unloading of the lattice and the conversion into kinetic energy occurs between Figure 5.6c and Figure 5.6d for case 3A, and between Figure 5.7c and Figure 5.7d for case 3B. One can see from those frames that the lattice at $t \approx 187.5$ μs has large deformations, meaning W^*_{int} is large, while the final frames show that most of the lattice is no longer as deformed, meaning W^*_{int} is lower. In the latter case, $E^*_{kin_{FEM}}$ is larger because the lattice is actually vibrating and the struts are currently moving in the opposite (the negative z-direction). Keep in mind that the simulations are stopped whenever the z-position of the centre of the impactor satisfies $z > T + r_{\text{impactor}}$ with T the original lattice thickness and r_{impactor} the impactor radius. This means that the simulation is stopped while the lattice is still vibrating. If one were to simulate for a longer time span, then one would see that W^*_{int} and $E^*_{kin_{FEM}}$ oscillate.

The last major difference compared to Figure 5.13 and Figure 5.15 is that initially, the energy absorption of the self-contact case is on-par or even better than the no self-contact case. One can see this in the form of $E^*_{kin_{impactor}}$ and $E^*_{mech_{FEM}}$. Before $t \approx 0.13$ ms, $E^*_{kin_{impactor}}$ of case 3A is actually lower than that of case 3B, implying more energy is absorbed in the beam-contact case. Similarly, for $t < 0.13$ ms, $E^*_{mech_{FEM}}$ is larger for case 3A compared to case 3B. This is also in line with what can be observed from Figure 5.10b. For $t < 0.13$ ms, the impactor z-velocity is decelerated equally or more by case 3A compared to case 3B. It is after this point in time that the no beam-contact case starts performing better in terms of energy absorption.

To explain this observed difference in behaviour compared to previous test cases, it is more convenient to look at the semi-logarithmic versions of the energy time histories. These have been included in Figure 5.18 below.



(a) The time history of the energy components of the system for case 3A at an initial velocity $v_0 = 150.0$ m/s with logarithmic scaling of the y-axis.

(b) The time history of the energy components of the system for case 3B at an initial velocity $v_0 = 150.0$ m/s with logarithmic scaling of the y-axis.

Figure 5.18: The time history of the energy components of the system at initial velocity $v_0 = 150.0$ m/s for cases 3A and 3B. A means with beam contact and B means without. All energy components are normalised to the initial impactor kinetic energy. The y-axis uses a logarithmic scale to make the small energy components better distinguishable. Note that a lower limit on the y-axis of 10^{-6} was chosen to keep the plot as a whole still readable.

First, Figure 5.18 has similarities with Figure 5.14 in that one can see that for the self-contact case, W^*_{frac} rises above 10^{-6} earlier than for the no self-contact case. This was also observed in Figure 5.14, where it was interpreted as a symptom of the fragmentation-induced fracture phenomenon discussed in Subsection 5.2.2, possibly causing the energy absorption performance of case 1A to be worse than case 1B. This phenomenon could also explain why the energy absorption performance of case 3A deteriorates relative to case 3B after $t \approx 0.13$ ms.

Second, Figure 5.18 shares a commonality with Figure 5.16: the kinetic and internal energy of the lattice

rise much earlier for case 3A than for case 3B. As explained for Figure 5.16, this could be an indication of compaction being captured by the computational model. The difference between Figure 5.18a and Figure 5.18b is not as large as for Figure 5.16, but that is likely because case 2A and case 2B use an octet lattice, which has more potential for contact events and thus the densification behaviour will be more prevalent.

Still, if case 3A indeed has managed to capture compaction initially, then that would explain why the energy absorption performance of the beam-contact case initially equalled or even exceeded the energy absorption performance of the no beam-contact case (Figure 5.10b). Then, after $t \approx 0.13$ ms, it is possible that the fragmentation-induced fracture of pristine struts downstream of the impactor degraded the energy absorption ability of the lattice, resulting in the lower deceleration of the impactor observed in Figure 5.10b for case 3A compared to case 3B past this point.

Energy absorption
In addition to the time histories of the energy components of the system for each test case discussed above, the energy absorption behaviour of each test case was quantified and plotted in Figure 5.19 to compare their relative energy absorption performance. The energy absorption measure plotted is the energy absorption efficiency, ΔW^*.

The absorbed energy efficiency is calculated using Equation 5.3. It is based on the mass-normalised energy absorption measure also used in [12] and [6].

$$\Delta W^* = \frac{\Delta W}{\bar{\rho}\rho\pi r_{\text{impactor}}^2 T} \tag{5.3}$$

In the equation above, $\bar{\rho}$ is the relative volume density of the unit cell of the lattice, ρ is the mass density of the lattice material, r_{impactor} is the radius of the impactor and T is the thickness of the lattice. The denominator describes the mass of the material directly in the path of the impactor, following Figure 2.19D. Lastly, ΔW is the total absorbed energy.

The total absorbed energy is calculated using Equation 5.4.

$$\Delta W = \frac{1}{2} m_{\text{impactor}} \left(v_0^2 - v_f^2 \right) \tag{5.4}$$

Here, m_{impactor} is the mass of the impactor, v_0 is the initial velocity of the impactor and v_f is the final velocity of the impactor.

Figure 5.19 shows the energy absorption efficiency, ΔW^* against v_0 for each test case.

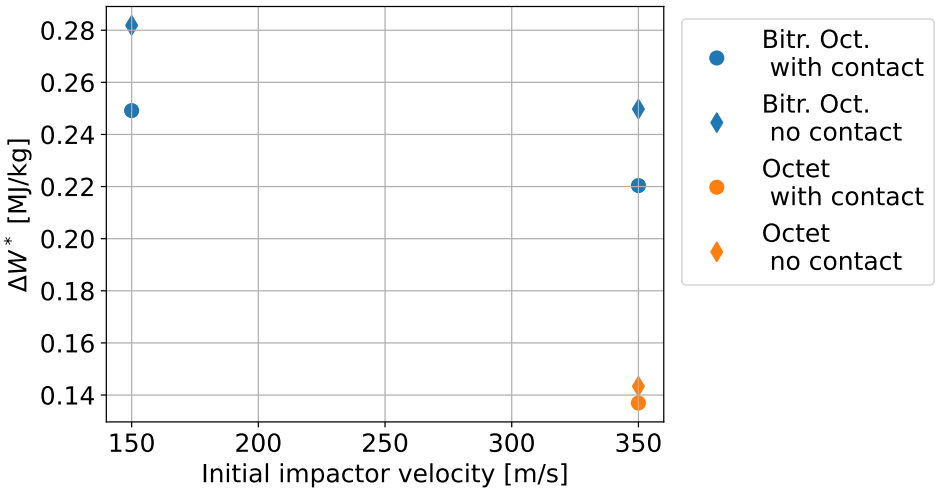


Figure 5.19: The mass-normalised energy absorption ΔW^* for each case for an initial velocity v_0 = 350.0 m/s and v_0 = 150.0 m/s.

Most notable from Figure 5.19 is a lack of data points, which makes it difficult to observe trends in the energy absorption performance for different values of initial impactor velocity, or, in the case of the octet lattices, entirely impossible. This is also why it is highly recommended for future work to conduct this large-scale impact simulation with more values of initial impactor velocities.

Still, the primary purpose of this numerical experiment was to evaluate the effect of self-contact on the energy absorption performance, and to this end, Figure 5.19 does offer some insights. For starters, the plot is congruent with Figure 5.10. For each level of initial impactor velocity and for each lattice type, the case with self-contact had worse energy absorption performance than the case without self-contact.

One can also see in Figure 5.19 that the disparity in energy absorption performance between the octet lattice with and without beam contact is smaller compared to the disparity in energy absorption performance between the bitruncated octahedron lattices with and without beam contact. As mentioned in Subsection 5.2.3, this may be the result of compaction not being as dominant of an absorption mechanism for octet lattices compared to bitruncated octahedron lattices. Since it has been established that proper compaction cannot be modelled by the current contact model if the simulation is run on multiple CPUs, the lack of proper compaction may have affected the performance of the octet lattice less than the bitruncated octahedron lattices.

To put the energy absorption performance in numbers, Table 5.5 summarises the mass-normalised energy absorption for each test case.

Table 5.5: A summary of the energy absorption performance of each test case. The third column lists the relative difference in mass-normalised energy absorption performance between a case with self-contact (A) and its associated case without (B).

Test case	ΔW^* [MJ/kg]	$\frac{(\Delta W^*)_A}{(\Delta W^*)_B} - 1$
1A	0.22037	-0.118
1B	0.24974	0.0
2A	0.13693	-0.045
2B	0.14335	0.0
3A	0.24915	-0.116
3B	0.28186	0.0

Table 5.5 can be used to calculate the relative difference in performance of the test cases with beam contact to the test cases without beam contact. This has been displayed in the third column of Table 5.5. It turns out that the bitruncated octahedron lattices with self-contact exhibit a near 12% loss in mass-normalised energy absorption performance compared to the bitruncated octahedron lattices without self-contact. This holds for both $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s. However, the lack of additional data points for other impactor initial velocities makes it difficult to say whether this performance difference is roughly constant for all v_0.

In the case of the octet lattices, the loss in mass-normalised energy absorption performance for the case with beam contact is approximately 4.5% compared to the case without beam contact. Because there is only one data point for the octet lattices, that at $v_0 = 350.0$ m/s, it is impossible to say what the relative performance difference would be over different values of v_0.

5.2.4. Summary of results

In summary, this chapter describes a large-scale numerical experiment of impact in truss metamaterials. The problem is modelled with a rigid spherical impactor that collides with the lattice. The aim was to evaluate the effect of self-contact on the energy absorption performance of the lattice. To this end, six test cases were simulated: three with self-contact and three without. Two lattice types were investigated, those being the bitruncated octahedron and the octet, but only the bitruncated octahedron lattices were able to be simulated at $v_0 = 350.0$ m/s and $v_0 = 150.0$ m/s, while the octet lattices could only be run with $v_0 = 350.0$ m/s due to time constraints.

For each test case, the deformed state was visualised in Paraview, the impactor trajectories and energy time histories were plotted, and the energy absorption was quantified. The impactor trajectories and the energy plots indicate that the test cases which account for contact between struts have worse energy absorption performance compared to the test cases which do not account for contact between struts. For $v_0 = 350.0$ m/s, the bitruncated octahedron lattice with self-contact performs roughly 11.8% worse than the same lattice without self-contact in terms of mass-normalised energy absorption. Similarly, for $v_0 = 150.0$ m/s, there is a loss of about 11.6% in mass-normalised energy absorption between the bitruncated octahedron lattice with self-contact and the same lattice without self-contact. In the case of the octet lattice, the relative difference in mass-normalised energy absorption for $v_0 = 350.0$ m/s of the case with self-contact is approximately 4.5% compared to the case without self-contact.

The results are contrary to the expectation that modelling contact between struts improves the energy absorption performance of a lattice. Firstly, modelling self-contact would allow the beam-contact-fracture framework to capture compaction behaviour in the lattice. As explained in Subsection 2.3.4, compaction is the stiffening effect of the lattice when it is compressed, and it was expected that this would improve impact performance. Secondly, modelling self-contact should also enable the lattice to spread the kinetic energy imparted by the impactor onto the lattice over a larger region due to kinetic energy transfer between contacting struts.

Possible explanations for why the energy absorption performance was worse when self-contact was accounted for come in two forms. The first is that compaction cannot be captured fully as a result of the relative small partitions. The simulations are run on multiple CPUs, and a limitation of the contact model is that inter-partition contact cannot be resolved, and will thus be missed. Without inter-partition contact, it becomes more difficult to take compaction into account. This makes for an important recommendation for future work. The second explanation is a hypothesised fragmentation-induced fracture phenomenon, which poses that during impact, the impactor causes fragments to be propelled downstream into intact struts. Their contact interactions can then cause failure of these struts, weakening the lattice ahead of the impactor and thus reducing the ability of the lattice to absorb energy.

6

Conclusion

6.1. Summary of findings

In summary, this thesis was dedicated to developing high-fidelity dynamic impact simulations of beam lattices capturing both strut self-contact and fracture behaviour. To connect with the research tasks posed in Chapter 1, the goals of this thesis were to (1) incorporate beam-contact behaviour with beam-fracture behaviour, and tie it with the strut dynamics; (2) test the functionality of the unified beam-contact-fracture framework in simple numerical experiments; (3) apply the unified beam-contact-fracture framework to a large-scale numerical model of impact in a truss metamaterial and evaluate the effect of self-contact on the energy absorption.

The process of integrating beam-contact behaviour with beam-fracture behaviour was covered in Chapter 3. Then, in the numerical experiments described in Chapter 4, the beam-contact-fracture framework showed it was able to maintain energy balance provided the time step Δt and the contact penalty stiffness were correctly tuned for the impact conditions. For this reason, a sensitivity study was performed on the contact penalty stiffness and a convergence study was conducted where Δt was reduced. The contact penalty stiffness sensitivity study showed that for a given Δt, there is a safe range of contact penalty stiffness values for which the energy balance is satisfied to a tolerance. The convergence study conducted at a fixed contact penalty stiffness and initial impact velocity revealed that energy response had already converged, but it is recommended for future work to perform the contact penalty sensitivity study for different values of Δt to verify whether Δt affects the aforementioned safe range of contact penalty stiffness values.

Finally, in Chapter 5, the unified beam-contact-fracture framework was demonstrated on a large-scale truss-metamaterial impact problem. Six test cases were simulated with the objective of evaluating the effect of self-contact on the energy absorption performance. Three test cases take beam-to-beam contact into account while the other three do not. Two lattice types were used in the simulations: the bitruncated octahedron and the octet. Due to time constraints, the bitruncated octahedron lattices were only simulated with initial impactor velocities of 350.0 m/s and 150.0 m/s while the octet lattices were only simulated with an initial impactor velocity of 350.0 m/s.

When the impact response of each test case was visualised, it was observed that each test case was perforated by the impactor. After the energy absorption was quantified for each test case, it turned out that self-contact degraded the energy absorption performance of the lattice. For the bitruncated octahedron lattices, the loss in mass-normalised absorbed energy when self-contact was modelled compared to when it was not, was approximately 12% for both initial impactor velocities. In the case of the octet lattices, the loss was about 4.5%. These results were contrary to expectations since self-contact should allow for capturing, among other phenomena, compaction. However, complete compaction is impossible to model for the beam-contact-fracture framework in the case of parallel computing (as was the case for the experiment in Chapter 5) because the contact model cannot resolve inter-partition contact. As such, it is strongly recommended to implement a fully parallelised contact detection routine.

6.2. Recommendations for future work

Over the course of performing the simple tests in Chapter 4, and conducting the main simulations in Chapter 5, some key suggestions have been formulated for any further developments of the beam-contact-fracture framework. In addition, Chapter 3 eluded to some major areas of improvement which future iterations of this framework ought to incorporate. The recommendations following below aim to either improve the accuracy and performance or expand the scope of the studied computational framework.

6.2.1. Convergence study of the self-contact penalty parameter

The effect of the contact penalty parameter used by the beam-contact model on the response has already been discussed in Subsection 4.3.2 and in Section 4.5. A simple sensitivity study was performed in Subsection 4.3.2 for a fixed Δt and v_0, which revealed that there is a safe range of contact penalty stiffness values for which the response of a system is within tolerance. However, choosing a contact penalty too low or too high causes the system response to diverge too much.

Despite the knowledge that the contact penalty parameter can greatly influence the results, time constraints have prevented a structured convergence analysis of the penalty parameter for every numerical experiment. Instead, the penalty parameter was simply tuned along with the time step size until contact was no longer missed and until the energy balance of the system was satisfactorily maintained.

It stands to reason that the performance of the computational framework and the efficiency of the entire workflow can only benefit from conducting a more organised convergence analysis of the self-contact penalty parameter. Such a convergence analysis would expand on the studies conducted in Subsection 4.3.2 and Subsection 4.3.3 by combining the two studies and investigating more parameters. For starters, the convergence study should run the contact penalty sensitivity study outlined in Subsection 4.3.2 for multiple values of Δt. This would establish the "safe" range of contact penalty values for a given Δt, turning the tuning process of the contact penalty value into a matter of looking up what the lower and upper bounds on the contact penalty stiffness are for a given Δt. This can then be further expanded to also including the initial impact velocity as an input to the convergence study. This way, one could obtain the safe range of contact penalty values for a given Δt and a given impact velocity.

This can be further supplemented by introducing and investigating a scaling law for the self-contact penalty parameter as a function of e.g. the impact velocity, so that one need not manually retune the penalty parameter for each simulation when the initial conditions are different. Doing so would save a lot of time otherwise lost to simulation runs where the contact model failed to detect contact events as a result of incorrect penalty values. Formulating a scaling law furthermore reduces the time spent on recalibrating the contact model for each different initial velocity level since it would only require calibrating the model once. However, this does front-load the tuning work to during the development and verification of the penalty scaling law.

6.2.2. Alternative to nodal kinetic energy

Subsection 3.3.2 mentioned an alternative to computing the kinetic energy of the lattice in the form of integrating the kinetic energy per unit length function, $\tilde{\Pi_{\text{kin}}}$, over the length of each element and then summing over each element (see Equation 3.74). This method of computing the system kinetic energy has already been implemented in SUMMIT for torsion-free geometrically nonlinear Kirchhoff rods, but in this thesis, the nodal kinetic energy defined by Equation 3.73 was used to compute the system kinetic energy. This method makes use of the lumped mass matrix and the nodal velocity field to easily compute the kinetic energy without requiring any integration over the length of the elements and subsequent summation over all elements. Because of this, the integrated kinetic energy approach was not used. Still, it may be worthwhile to rerun the numerical experiments from Chapter 4 with the integrated kinetic energy to see how much of a difference integrating and summing over the elements makes in terms of maintaining energy balance.

6.2.3. Direct nodal residual contribution of contact point-forces

As mentioned earlier in Subsection 3.1.7, a more refined alternative to distributing the contact point-force to the nearest quadrature points is computing the contributions of the singularity force directly to the nodal residual force. By making use of the sifting property of the Dirac pulse and the fact that the element shape functions are known at the point of contact, the equivalent forces and moments at the nodes of the element can be calculated from the point force. These nodal forces and moments can then be added directly to the residual force nodal field, thereby better preserving the fact that the contact force is a point force as opposed to a pseudo-distributed force when it is distributed over the nearest quadrature points. However, special care is required when one also intends to implement a line-to-line contact formulation in addition to the point-to-point contact formulation (see Subsection 6.2.6) because the contact force for line-contact is a distributed force.

6.2.4. Inter-partition contact detection

One of the key assumptions made in Section 3.2 is the fact that inter-partition contact events are not tracked. This is a physical limitation of the implementation of the contact model. Because each processor only knows of the elements in its own partition, it can only detect contact events happening between elements within its own partition. For any processor to detect contact between elements of any partition, a major modification of the contact detection routine will be necessary.

For starters, the nearest neighbour search subroutine described in Subsection 3.1.7 must be modified to be compatible with parallelisation over multiple partitions. Having each partition run this algorithm for all elements in all the other partitions is not efficient. Possible solutions to fully parallelise the contact algorithm depend on how complete the capture of all contact events must be. If one is only interested in capturing all contact cases near each partition, then one could implement a method for spatially nearby partitions to handle inter-partition contact. The contact detection routine then consists of an intra-partition part, which handles contact as described in Subsection 3.1.7 and an inter-partition part which searches for possible contact pairs between the closest partitions.

The limitation of this approach is that it would only be computationally efficient when the inter-partition contact process is run for a set of known nearby partitions, rather than searching over all partitions. This immediately precludes the detection of contact between elements from partitions which are spatially far away from one another, such that they are not considered neighbouring partitions. The relevance of this limitation becomes apparent when one considers that the collision with the spherical impactor in Chapter 5 causes the affected struts to fracture and fragment, shooting in multiple directions and thus potentially interacting and colliding with the surrounding structure. It is under these circumstances that fragments from partitions close to the impact zone could very well find themselves near the struts of cells far away from the impact zone. The abovementioned nearest partitions inter-partition contact method would not be able to detect these collisions.

Detecting contact anywhere in the lattice would require more severe modifications to the model. Perhaps one could choose to use dedicated processors for handling contact. However, when these processors do not perform the same tasks as the processors allocated to handling the rest of the simulation, then synchronising them becomes a challenge. One should hopefully notice that the implementation of a fully parallel contact routine is not trivial and deserves its own dedicated study. Still, this is a worthwhile endeavour as Chapter 5 identified the incomplete capture of compaction in the case of parallel computing as a possible reason why the energy absorption results were not as expected. Implementing a fully parallel contact model would allow one to rerun the same experiment from Chapter 5 and hopefully capture compaction properly.

6.2.5. Alternatives to penalty contact

The contact enforcement technique employed by the contact model studied in this thesis is based on the penalty method. Subsection 2.6.6 explains why the penalty approach is preferred for beam-contact problems over the other commonly used constraint enforcement technique, the Lagrange multiplier method. The biggest reason to prefer the penalty method is the fact that it does not introduce extra unknowns, thereby saving on computational costs compared to the Lagrange multiplier method. However, Chapter 4 has made it apparent that tuning the penalty parameter is a time consuming process.

Using penalty contact means finding a delicate balance between two undesirable outcomes: too low a penalty parameter and the contact event is missed entirely; too high a penalty parameter and the system becomes overly stiff and possibly unstable.

In light of this, it may be worthwhile to investigate alternative approaches to enforcing the contact constraint which do not involve a penalty parameter. The Lagrange multiplier method is probably the foremost example of a penalty-free enforcement technique. Despite its higher computational cost, the extra time required for time marching may well be offset by the fact that no time will be lost tuning a penalty parameter since constraint enforcement with the Lagrange multiplier method does not rely on a penalty parameter.

Another promising alternative to the penalty method is the Decomposition Contact Response (DCR) method. Raso (2024) dedicated his Master's thesis to this penalty-free contact enforcement technique [48]. In short, DCR utilises conservation of momentum and the fact that the pre-impact momenta are known to calculate the post-impact momenta. After determining the post-impact velocities, any detected penetration is removed through some form of closest-point projection [106] [48]. These two steps combine to result in the post-impact state. One will notice that the DCR method requires no contact penalty parameter, which means no penalty parameter tuning. Thanks to Raso's work, the DCR method has been implemented in SUMMIT for contact between continuous Galerkin beam elements. As such, there is an opportunity for a follow-up study to modify the DCR method to be compatible with the discontinuous Galerkin formulation so that a penalty-free beam-contact-fracture framework can be developed.

Of course, making the DCR method work with the discontinuous Galerkin formulation introduces challenges of its own, such as implementing the closest-point projection for elements which do not share nodes. In the DG formulation, elements generally do not share nodes, so translating the nodes of one element during penetration removal will not automatically do the same for any neighbouring elements, potentially causing displacement discontinuities at their interfaces. One would need to keep track of which elements share interface boundaries to correctly perform the intersection removal after contact.

6.2.6. Implementation of line-to-line contact

Subsection 2.6.6 also mentions two different formulations for modelling beam-to-beam contact: point-to-point contact and line-to-line contact. The beam-contact-fracture framework studied in this thesis uses the point-contact formulation, as explained in Subsection 3.1.5. The simplicity of the point-contact model made it more attractive than the more robust line-contact model, especially for an initial attempt to integrate contact with beam fracture. However, it is undeniable that the line-contact formulation has a higher consistency and robustness than the point-contact formulation, as explained by Meier et al. (2016) [17] and summarised in Subsection 2.6.6.

With this in mind, it could be highly beneficial long-term to implement the line-to-line contact model proposed by Meier et al. (2016) in SUMMIT, especially considering that the validity for arbitrary contact angles would make this contact formulation well-suited to problems where wound slender structures are present, such as filament wind simulations. The point-contact model performs well for impact problems where unique contact points are not difficult to find, but it would likely suffer in winding problems because of the small contact angles (as illustrated by (c) of Figure 2.40).

The caveat is that the line-to-line contact model is not as computationally efficient as the point-to-point contact model as mentioned in Subsection 2.6.6 and Subsection 3.1.5. However, Subsection 3.1.5 mentions a solution to this issue in the form of a hybrid contact model which combines point-contact and line-contact. Meier et al. (2017) proposed this formulation to get the best of both worlds since it features the robustness of the line-to-line formulation for small contact angles while also benefiting from the computational efficiency of point-to-point contact for large contact angles [95]. As such, if one would like to implement a line-to-line contact model, then one may as well implement the hybrid contact formulation since the point-contact formulation has already been implemented in SUMMIT.

6.2.7. Accounting for more impact phenomena

Subsection 2.3.4 mentioned six phenomena found in the specimen when subjected to impact: plasticity, fracture, shock waves, thermal effects, kinetic energy transfer and densification. As outlined in Section 3.3, the beam-contact-fracture framework only considers fracture and contact, and no plasticity or thermal effects. Furthermore, Section 3.1 explains that the framework utilises a hyperelastic constitutive model and assumes frictionless contact. As a result, the only phenomena which can be captured by the current iteration of the computational framework are fracture, kinetic energy transfer and densification, since these only require fracture and contact models. The rest of the possible physical phenomena necessitate additional models to supplement the fracture and contact models.

Friction

Friction was not listed explicitly as an absorption mechanism in Subsection 2.3.4, although it could be understood to be a part of kinetic energy transfer and thermal effects. The fact is that some kinetic energy should be lost during contact because of friction, and the current contact model does not reflect this. While it is true that a frictional contact force model has been implemented in SUMMIT, the computational experiments conducted in this thesis did not make use of frictional contact for simplicity as explained in Subsection 3.1.7. Modelling the tangential contact forces requires providing a friction coefficient, which necessitates determining said friction coefficient for each material. Formally speaking, this coefficient would need to be determined empirically. Still, the particularities of determining the friction coefficient aside, it would be a clear improvement to the accuracy of the contact model to account for friction too.

Plasticity

One of the major physical phenomena in impact problems not accounted for by the current impact model is plasticity. The material model assumed by the beam-contact-fracture framework is hyperelastic, however, Section 2.3 makes it clear that even at lower impact velocities, plastic effects can be observed. These plastic effects include strain hardening and permanent deformations. Modelling plastic effects would improve the accuracy of the computational impact framework, and it is a natural next step for the development of an impact model for low to moderately high impact velocities. In fact, integrating plasticity into beam mechanics is a topic of active research and development in Dr Giovanardi's research group.

Hydrodynamics

Friction and plasticity are present already at low to moderately high impact velocities, but shock waves and thermal effects are usually only observed at high impact velocities. The former of the two phenomena are governed by a hydrodynamic model. Also, as mentioned in Subsection 2.3.3, the material starts exhibiting fluid-like behaviour at high impact velocities, which also necessitates a hydrodynamic model. Unlike friction and plasticity models, which supplement the impact model, but are not necessary to model impact at lower velocities, a hydrodynamic model is essentially a requirement for accurately modelling impact at high velocities because the dominant physical phenomena present in high velocity impact are flow dominated rather than material strength dominated. As such, if one would like to extend the computational framework to the high velocity impact regime, which is a highly relevant impact regime for impact-protection of spacecraft, then developing a hydrodynamic model is the first step. One would both need an equation of state to describe the material's constitutive behaviour in the high velocity impact regime, and governing equations for the shock waves that occur in high velocity impact problems.

Thermomechanics

Closely tied with hydrodynamics is modelling the thermal effects present in high velocity impact problems. These thermal effects could arise from the shock waves themselves, shock heating, or from the heat generated during deformation, such as plastic heating (see Subsection 2.3.3). Modelling the effect of temperature on the deformation requires developing governing equations describing the change in temperature as a result of a change in energy (be it due to shock waves or due to plastic heating), and it requires a temperature-dependent constitutive law. Together, they describe the thermomechanical response of the struts. Similar to the hydrodynamic model, modelling the thermomechanical response of the material during high velocity impact is effectively a necessity for modelling accurate impact

responses, so it is highly recommended to develop a thermomechanical model for the computational impact framework if the intent is to eventually simulate high velocity impact.

6.2.8. Non-slender beam formulations

The beam formulation used in the computational model is the torsion-free geometrically nonlinear Kirchhoff beam formulation. The validity of this beam model is dependent on how slender the structure is. Because these beam elements can only be used to model slender structures, it places a limit on the relative density $\bar{\rho}$ of the unit cells. If $\bar{\rho}$ becomes too large, then the strut radius becomes too large for it to be considered slender. At this point, the assumptions underpinning the torsion-free formulation of the geometrically nonlinear Kirchhoff beam, and even the shear-free assumption of the Kirchhoff beam may become invalid. This upper limit of $\bar{\rho}$ is a physical limitation of the beam formulation, and modelling lattices with higher values of $\bar{\rho}$, such as the relative densities of the lattices used by Portela et al. (2021) [64], requires using a different beam formulation altogether.

The obvious alternative to the torsion-free geometrically nonlinear Kirchhoff beam for non-slender beams would the Simo-Reissner beam. As can be seen in Subsection 2.5.3, this formulation does not make assumptions regarding the shear-stiffness (or torsion-stiffness) of the beam. As such, modelling denser lattice structures can be achieved by implementing a Simo-Reissner beam element formulation. Of course, compared to the torsion-free geometrically nonlinear Kirchhoff beam formulation, a Simo-Reisser beam element introduces additional unknowns which increases the computational load.

Another beam formulation which would also work for less slender beams, and which is currently in development in Dr Giovanardi's research group, is a shear-stiff corotational beam model. The corotational formulation of an element allows for separating rigid-body displacements from the stretch tensor, resulting in computationally efficient and robust elements [107]. Because this beam formulation has already been implemented in SUMMIT, the focus can shift more towards integrating this element type with the beam-contact-fracture framework. In principle, the computational model is not limited to torsion-free geometrically nonlinear Kirchhoff beam elements, but integrating a new element type would require a proper investigation to see whether the code runs without issue when a different element type is used. In any case, making the framework compatible with the corotational beam formulation is highly recommended to model lattices with higher values of $\bar{\rho}$.

6.2.9. Preventing boundary effects

In the impact response visualisations from Chapter 5, there seemed to be indications of boundary effects for $v_0 = 150.0$ m/s. The deformation waves in some of the captured frames looked like they reached all the way to the boundaries. This means that the lattice dimensions of 50.0 x 50.0 x 25.0 mm may have been too small. To prevent boundary effects, the lattice should have been larger in x and y. It may be worthwhile to rerun the large-scale truss-metamaterial impact simulations from Chapter 5 with a larger lattice, 75.0 x 75.0 x 25.0 mm or even 100.0 x 100.0 x 25.0 mm in dimensions.

6.2.10. Expanding the scope of the impact tests

The last recommendations pertain to expanding the scope of the simulated impact tests themselves, rather than improving upon the computational model. Because of time constraints, the large-scale lattice impact simulation was only run with one type of material, one value of relative density, and two initial impactor velocities. This means that there is the potential of a follow-up study into the impact performance of different materials and different lattice geometries for a larger range of impactor velocities. As mentioned in Chapter 5, the impact problem already considers three possible materials, but this study only used one material. Therefore, it is already possible to run simulations for different materials and compare the impact performance of each specimen.

The other recommendation which is also already possible to implement, is running the simulations at more values of the initial impactor velocity. As can be seen in Figure 5.19, having more data points for more initial velocities would allow one to more easily and accurately predict trends in the energy absorption behaviour. As it stands now, having just two or fewer data points for each lattice type in the energy absorption plots makes it difficult or impossible to identify and motivate trends. It would therefore be a great idea to run the simulations at more initial impactor velocities. The only limitation

is time, which is the reason why this study only managed to run the simulations for v_0 = 350.0 m/s and v_0 = 150.0 m/s, but aside from time, there is nothing stopping one from running the simulations at more levels of initial impactor velocities, as long as v_0 does not become so high that hydrodynamic and thermal effects should be modelled.

As for lattice geometries, the impact application was designed for two cell topologies: the octet and the bitruncated octahedron. Modelling impact in lattices consisting of different cell topologies like the rhombic dodecahedron or the BFCC as Bieler and Weinberg (2025) [6] have done, would require modifying the driver of the application to also handle the different cell topologies. Aside from cell topology, relative density is also a key input. Strictly speaking, users can simply define their relative density of choice, and the mesh of the lattice will be generated with that relative density. Chapter 5 only used a relative density of 1.25%, but one could also run the simulations with lattices with $\bar{\rho}$ up to 5.0%.

However, increasing the value of $\bar{\rho}$ means the struts become less slender, so as explained above in Subsection 6.2.8, there is an upper limit to $\bar{\rho}$, beyond which the results will not be valid if torsion-free geometrically nonlinear Kirchhoff beam elements are used. A proper comparison of the effect of relative lattice density on impact performance would require a larger range of $\bar{\rho}$ (see the range of $\bar{\rho}$ used by Bieler and Weinberg (2025) [6] and Subsection 2.2.3) than is currently possible. This ties in with Subsection 6.2.8, meaning that incorporating a non-slender beam model into the computational framework is a prerequisite for expanding the scope of the impact tests to cover a larger range of relative densities. Doing so would allow one to find out which lattice configuration is the most efficient for energy absorption.

7

Declaration of generative AI and AI tools

No AI tools were used during the writing of this thesis and the code development.

References

[1] J. Sun and J. Zhou. "Metamaterials: The Art in Materials Science". *Engineering* 44 (2025), pp. 145–161. DOI: https://doi.org/10.1016/j.eng.2024.12.011.

[2] T. A. Schaedler and W. B. Carter. "Architected Cellular Materials". *Annual Review of Materials Research* 46.Volume 46, 2016 (July 2016), pp. 187–210. DOI: 10.1146/ANNUREV-MATSCI-070115-031624/CITE/REFWORKS.

[3] A. G. Evans, M. Y. He, V. S. Deshpande, J. W. Hutchinson, A. J. Jacobsen, and W. B. Carter. "Concepts for enhanced energy absorption using hollow micro-lattices". *International Journal of Impact Engineering* 37 (9 Sept. 2010), pp. 947–959. DOI: 10.1016/j.ijimpeng.2010.03.007.

[4] E. S. D. Office. *ESA's ANNUAL SPACE ENVIRONMENT REPORT*. Technical Report. ESA ESOC, 2025.

[5] A. Nešić, O. Cokorilo, and S. Steiner. "Aircraft Repair and Withdrawal Costs Generated by Bird Collision with the Windshield". *Promet - Traffic & Transportation* 29 (Dec. 2017), pp. 623–629. DOI: 10.7307/ptt.v29i6.2448.

[6] S. Bieler and K. Weinberg. "Energy absorption in lattice-structured materials under impact loading". *European Journal of Mechanics - A/Solids* 114 (2025), p. 105734. DOI: https://doi.org/10.1016/j.euromechsol.2025.105734.

[7] W. Jiang, G. Yin, L. Xie, and M. Yin. "Multifunctional 3D lattice metamaterials for vibration mitigation and energy absorption". *International Journal of Mechanical Sciences* 233 (2022), p. 107678. DOI: https://doi.org/10.1016/j.ijmecsci.2022.107678.

[8] W. G. Colvin T.J. Karcz J. "Cost and Benefit Analysis of Orbital Debris Remediation". *Nasa report* (Mar. 2023).

[9] C. Chen, Y. He, Y. Chen, G. Lu, M.-H. Lu, and X. Li. "Impact dynamics of mechanical metamaterials: A short review and perspective". *Forces in Mechanics* 21 (2025), p. 100335. DOI: https://doi.org/10.1016/j.finmec.2025.100335.

[10] J. Ayers, B. Chamish, T.-K. Chen, A. Ghoshal, C. Kittur, M. Coatney, N. Bradley, and T. Jones. *Auxetic Metamaterials under Direct Impact Loads in a Structural Health Monitoring Framework*. Tech. rep. ARL-TR-6306. Aberdeen Proving Ground, MD: U.S. Army Research Laboratory, 2013.

[11] D. Dattelbaum, A. Ionita, B. Patterson, B. Branch, and L. Kuettner. "Shockwave dissipation by interface-dominated porous structures". *AIP Advances* 10 (July 2020), p. 075016. DOI: 10.1063/5.0015179.

[12] T. Butruille, J. C. Crone, and C. M. Portela. "Decoupling particle-impact dissipation mechanisms in 3D architected materials". *Proceedings of the National Academy of Sciences* 121.6 (2024), e2313962121. DOI: 10.1073/pnas.2313962121. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2313962121.

[13] S. Gaitanaros and S. Kyriakides. "Dynamic crushing of aluminum foams: Part II – Analysis". *International Journal of Solids and Structures* 51.9 (2014), pp. 1646–1661. DOI: https://doi.org/10.1016/j.ijsolstr.2013.11.020.

[14] J. Mueller, K. H. Matlack, K. Shea, and C. Daraio. "Energy Absorption Properties of Periodic and Stochastic 3D Lattice Materials". *Advanced Theory and Simulations* 2.10 (2019), p. 1900081. DOI: https://doi-org.tudelft.idm.oclc.org/10.1002/adts.201900081. eprint: https://advanced-onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/pdf/10.1002/adts.201900081.

[15] J. C. Crone, J. Knap, and R. Becker. "Multiscale modeling of 3D nano-architected materials under large deformations". *International Journal of Solids and Structures* 252 (2022), p. 111839. DOI: https://doi.org/10.1016/j.ijsolstr.2022.111839.

[16] S. K. Kota, S. Kumar, and B. Giovanardi. "A discontinuous Galerkin/cohesive zone model approach for the computational modeling of fracture in geometrically exact slender beams". *Computational Mechanics* 75.2 (July 2024), pp. 595–612. DOI: 10.1007/s00466-024-02521-0.

[17] C. Meier, A. Popp, and W. Wall. "A finite element approach for the line-to-line contact interaction of thin beams with arbitrary orientation". 308 (2016), pp. 377–413. DOI: 10.1016/j.cma.2016.05.012.

[18] M. L. Diallo, Á. Sarankó, T. I. Pataki, and L. Zsidai. "Metamaterials and their applications in engineering metamaterials: Classification, Applications - A Comprehensive Review". *Applied Physics A* 132.3 (Feb. 2026). DOI: 10.1007/s00339-026-09309-4.

[19] R. Kumar, M. Kumar, J. S. Chohan, and S. Kumar. "Overview on metamaterial: History, types and applications". *Materials Today: Proceedings* 56 (2022). 3rd International Conference on Contemporary Advances in Mechanical Engineering, pp. 3016–3024. DOI: https://doi.org/10.1016/j.matpr.2021.11.423.

[20] M. F. Ashby and L. J. Gibson. "Cellular solids: structure and properties". *Press Syndicate of the University of Cambridge, Cambridge, UK* (1997), pp. 175–231.

[21] L. Bogunia and K. Weinberg. "Experimental and numerical investigation of PUR foam under dynamic loading". *Proceedings in Applied Mathematics and Mechanics* 16.1 (2016), pp. 309–310. DOI: https://doi.org/10.1002/pamm.201610143. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.201610143.

[22] R. Mines. *Metallic Microlattice Structures: Manufacture, Materials and Application*. Cham, Switzerland: Springer, 2019.

[23] S. Kokil-Shah, A. Sur, S. Darvekar, and M. Shah. "Recent advancements of micro-lattice structures: Application, manufacturing methods, mechanical properties, topologies and challenges". *Arabian Journal for Science and Engineering* 46.12 (Aug. 2021), pp. 11587–11600. DOI: 10.1007/s13369-021-05992-y.

[24] J. Xiong, R. Mines, R. Ghosh, A. Vaziri, L. Ma, A. Ohrndorf, H.-J. Christ, and L. Wu. "Advanced Micro-Lattice Materials". *Advanced Engineering Materials* 17.9 (2015), pp. 1253–1264. DOI: https://doi-org.tudelft.idm.oclc.org/10.1002/adem.201400471. eprint: https://advanced-onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/pdf/10.1002/adem.201400471.

[25] X. Zhang, Y. Wang, B. Ding, and X. Li. "Design, Fabrication, and Mechanics of 3D Micro-/Nanolattices". *Small* 16.15 (2020), p. 1902842. DOI: https://doi-org.tudelft.idm.oclc.org/10.1002/smll.201902842. eprint: https://onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/pdf/10.1002/smll.201902842.

[26] H. Yin, W. Zhang, L. Zhu, F. Meng, J. Liu, and G. Wen. "Review on lattice structures for energy absorption properties". *Composite Structures* 304 (2023), p. 116397. DOI: https://doi.org/10.1016/j.compstruct.2022.116397.

[27] S. Li, R. Yang, S. Sun, and B. Niu. "Advances in the analysis of honeycomb structures: A comprehensive review". *Composites Part B: Engineering* 296 (2025), p. 112208. DOI: https://doi.org/10.1016/j.compositesb.2025.112208.

[28] L. Hu, M. Zhou, and H. Deng. "Dynamic crushing response of auxetic honeycombs under large deformation: Theoretical analysis and numerical simulation". *Thin-Walled Structures* 131 (2018), pp. 373–384. DOI: https://doi.org/10.1016/j.tws.2018.04.020.

[29] A. Alderson, K. Alderson, D. Attard, K. Evans, R. Gatt, J. Grima, W. Miller, N. Ravirala, C. Smith, and K. Zied. "Elastic constants of 3-, 4- and 6-connected chiral and anti-chiral honeycombs subject to uniaxial in-plane loading". *Composites Science and Technology* 70.7 (2010). Special issue on Chiral Smart Honeycombs, pp. 1042–1048. DOI: https://doi.org/10.1016/j.compscitech.2009.07.009.

[30] J. N. Grima and K. E. Evans. "Auxetic behavior from rotating triangles". *Journal of materials science* 41.10 (2006), pp. 3193–3196.

[31] F. Sun, C. Lai, H. Fan, and D. Fang. "Crushing mechanism of hierarchical lattice structure". *Mechanics of Materials* 97 (2016), pp. 164–183. DOI: https://doi.org/10.1016/j.mechmat.2016.02.016.

[32] W. Zhang, S. Yin, T. Yu, and J. Xu. "Crushing resistance and energy absorption of pomelo peel inspired hierarchical honeycomb". *International Journal of Impact Engineering* 125 (2019), pp. 163–172. DOI: https://doi.org/10.1016/j.ijimpeng.2018.11.014.

[33] Q. Lu, D. Qi, Y. Li, D. Xiao, and W. Wu. "Impact energy absorption performances of ordinary and hierarchical chiral structures". *Thin-Walled Structures* 140 (2019), pp. 495–505. DOI: https://doi.org/10.1016/j.tws.2019.04.008.

[34] T. Tancogne-Dejean and D. Mohr. "Elastically-isotropic truss lattice materials of reduced plastic anisotropy". *International Journal of Solids and Structures* 138 (2018), pp. 24–39. DOI: https://doi.org/10.1016/j.ijsolstr.2017.12.025.

[35] I. Maskery, N. Aboulkhair, A. Aremu, C. Tuck, I. Ashcroft, R. Wildman, and R. Hague. "A mechanical property evaluation of graded density Al-Si10-Mg lattice structures manufactured by selective laser melting". *Materials Science and Engineering: A* 670 (2016), pp. 264–274. DOI: https://doi.org/10.1016/j.msea.2016.06.013.

[36] Z. Xiao, Y. Yang, R. Xiao, Y. Bai, C. Song, and D. Wang. "Evaluation of topology-optimized lattice structures manufactured via selective laser melting". *Materials & Design* 143 (2018), pp. 27–37. DOI: https://doi.org/10.1016/j.matdes.2018.01.023.

[37] B. Bahrami Babamiri, H. Askari, and K. Hazeli. "Deformation mechanisms and post-yielding behavior of additively manufactured lattice structures". *Materials & Design* 188 (2020), p. 108443. DOI: https://doi.org/10.1016/j.matdes.2019.108443.

[38] X. Yan, Q. Li, S. Yin, Z. Chen, R. Jenkins, C. Chen, J. Wang, W. Ma, R. Bolot, R. Lupoi, Z. Ren, H. Liao, and M. Liu. "Mechanical and in vitro study of an isotropic Ti6Al4V lattice structure fabricated using selective laser melting". *Journal of Alloys and Compounds* 782 (2019), pp. 209–223. DOI: https://doi.org/10.1016/j.jallcom.2018.12.220.

[39] T. Tancogne-Dejean, M. Diamantopoulou, M. B. Gorji, C. Bonatti, and D. Mohr. "3D Plate-Lattices: An Emerging Class of Low-Density Metamaterial Exhibiting Optimal Isotropic Stiffness". *Advanced Materials* 30.45 (2018), p. 1803334. DOI: https://doi.org/10.1002/adma.201803334. eprint: https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201803334.

[40] S. Wang, Z. Zheng, C. Zhu, Y. Ding, and J. Yu. "Crushing and densification of rapid prototyping polylactide foam: Meso-structural effect and a statistical constitutive model". *Mechanics of Materials* 127 (2018), pp. 65–76. DOI: https://doi.org/10.1016/j.mechmat.2018.09.003.

[41] S. Duan, W. Wen, and D. Fang. "Additively-manufactured anisotropic and isotropic 3D plate-lattice materials for enhanced mechanical performance: Simulations & experiments". *Acta Materialia* 199 (2020), pp. 397–412. DOI: https://doi.org/10.1016/j.actamat.2020.08.063.

[42] C. Bonatti and D. Mohr. "Smooth-shell metamaterials of cubic symmetry: Anisotropic elasticity, yield strength and specific energy absorption". *Acta Materialia* 164 (2019), pp. 301–321. DOI: https://doi.org/10.1016/j.actamat.2018.10.034.

[43] L. Zhang, Z. Hu, M. Y. Wang, and S. Feih. "Hierarchical sheet triply periodic minimal surface lattices: Design, geometric and mechanical performance". *Materials & Design* 209 (2021), p. 109931. DOI: https://doi.org/10.1016/j.matdes.2021.109931.

[44] X. Zheng, W. Smith, J. Jackson, B. Moran, H. Cui, D. Chen, J. Ye, N. Fang, N. Rodriguez, T. Weisgraber, and C. Spadaccini. "Multiscale Metallic Metamaterials". *Nature Materials* 15.10 (July 2016), pp. 1100–1106. DOI: 10.1038/nmat4694.

[45] S. Yin, H. Chen, J. Li, T. Yu, and J. Xu. "Effects of architecture level on mechanical properties of hierarchical lattice materials". *International Journal of Mechanical Sciences* 157-158 (2019), pp. 282–292. DOI: https://doi.org/10.1016/j.ijmecsci.2019.04.051.

[46] A. Osis. "Fracture and Impact Dynamics of Beam Lattices". MA thesis. Delft University of Technology, 2025.

[47] S. Demian. "Hybrid Beam-3D Continuum Nonlinear Modeling of Architected Truss Metamaterials". MA thesis. Delft University of Technology, 2026.

[48] D. C. Raso. "A momentum-based, penalty-free method for modelling contact between geometrically exact beams in finite element frameworks ". MA thesis. Delft University of Technology, 2024.

[49] J. Li and J. D. Lee. *Molecular simulations and multiphysics nanoscale coupling*. CRC Press, 2025.

[50] T. Tancogne-Dejean, A. B. Spierings, and D. Mohr. "Additively-manufactured metallic micro-lattice materials for high specific energy absorption under static and dynamic loading". *Acta Materialia* 116 (2016), pp. 14–28. DOI: https://doi.org/10.1016/j.actamat.2016.05.054.

[51] T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, and W. B. Carter. "Ultralight Metallic Microlattices". *Science* 334.6058 (2011), pp. 962–965. DOI: 10.1126/science.1211649. eprint: https://www.science.org/doi/pdf/10.1126/science.1211649.

[52] J. Bauer, S. Hengsbach, I. Tesari, R. Schwaiger, and O. Kraft. "High-strength cellular ceramic composites with 3D microarchitecture". *Proceedings of the National Academy of Sciences* 111.7 (2014), pp. 2453–2458. DOI: 10.1073/pnas.1315147111. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1315147111.

[53] Z. C. Eckel, C. Zhou, J. H. Martin, A. J. Jacobsen, W. B. Carter, and T. A. Schaedler. "Additive manufacturing of polymer-derived ceramics". *Science* 351.6268 (2016), pp. 58–62. DOI: 10.1126/science.aad2688. eprint: https://www.science.org/doi/pdf/10.1126/science.aad2688.

[54] T. George, V. S. Deshpande, and H. N. Wadley. "Mechanical response of carbon fiber composite sandwich panels with pyramidal truss cores". *Composites Part A: Applied Science and Manufacturing* 47 (2013), pp. 31–40. DOI: https://doi.org/10.1016/j.compositesa.2012.11.011.

[55] K. C. Cheung and N. Gershenfeld. "Reversibly Assembled Cellular Composite Materials". *Science* 341.6151 (2013), pp. 1219–1221. DOI: 10.1126/science.1240889. eprint: https://www.science.org/doi/pdf/10.1126/science.1240889.

[56] H. Wang, Y. Lyu, S. Bosiakov, H. Zhu, and Y. Ren. "A review on the mechanical metamaterials and their applications in the field of biomedical engineering". *Frontiers in Materials* Volume 10 - 2023 (2023). DOI: 10.3389/fmats.2023.1273961.

[57] J. U. Surjadi, L. Gao, H. Du, X. Li, X. Xiong, N. X. Fang, and Y. Lu. "Mechanical Metamaterials and Their Engineering Applications". *Advanced Engineering Materials* 21.3 (2019), p. 1800864. DOI: https://doi-org.tudelft.idm.oclc.org/10.1002/adem.201800864. eprint: https://advanced-onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/pdf/10.1002/adem.201800864.

[58] W. J. Stronge. *Impact mechanics*. Cambridge University Press, 2010.

[59] S. Yan, W. Liu, X. Tan, Z. Meng, W. Luo, H. Jin, Y. Wen, J. Sun, L. Wu, and J. Zhou. "Bio-inspired mechanical metamaterial with ultrahigh load-bearing capacity for energy dissipation". *Materials Today* 77 (2024), pp. 11–18. DOI: https://doi.org/10.1016/j.mattod.2024.06.003.

[60] R. Xu, Y. He, X. Li, M. Lu, and Y. Chen. "Snap-fit mechanical metamaterials". *Applied Materials Today* 30 (2023), p. 101714. DOI: https://doi.org/10.1016/j.apmt.2022.101714.

[61] M. Sood and C.-M. Wu. "Influence of structural arrangements on static and dynamic properties of additively manufactured polyester elastomer lattice metamaterials". *Applied Materials Today* 35 (2023), p. 101933. DOI: https://doi.org/10.1016/j.apmt.2023.101933.

[62] M. Pham, C. Liu, I. Todd, and J. Lertthanasarn. "Damage-tolerant architected materials inspired by crystal microstructure". *Nature* 565 (7739 Jan. 2019), pp. 305–311. DOI: 10.1038/s41586-018-0850-3.

[63] S. Watts. "Elastic response of hollow truss lattice micro-architectures". *International Journal of Solids and Structures* 206 (Dec. 2020), pp. 472–564. DOI: 10.1016/j.ijsolstr.2020.08.018.

[64] C. M. Portela, B. W. Edwards, D. Veysset, Y. Sun, K. A. Nelson, D. M. Kochmann, and J. R. Greer. "Supersonic impact resilience of nanoarchitected carbon". *Nature Materials 2021 20:11* 20.11 (June 2021), pp. 1491–1497. DOI: 10.1038/s41563-021-01033-z.

[65] H. A. Signetti S. "Transition regime between high-velocity and hypervelocity impact in metals – A review of the relevant phenomena for material modeling in ballistic impact studies". *International Journal of Impact Engineering* 167 (Sept. 2022). DOI: 10.1016/j.ijimpeng.2022.104213.

[66] M. E. Backman and W. Goldsmith. "The mechanics of penetration of projectiles into targets". *International Journal of Engineering Science* 16.1 (1978), pp. 1–99. DOI: https://doi.org/10.1016/0020-7225(78)90002-2.

[67] C. E. Anderson and S. R. Bodner. "Ballistic impact: The status of analytical and numerical modeling". *International Journal of Impact Engineering* 7.1 (1988), pp. 9–35. DOI: https://doi.org/10.1016/0734-743X(88)90010-3.

[68] J. Zukas. *Introduction to Hydrocodes*. Baltimore, USA: Computational Mechanics Association, 2004.

[69] C. Meier, A. Popp, and W. Wall. "Geometrically exact finite element formulations for slender beams: Kirchhoff-Love theory versus Simo-Reissner theory". *Archives of Computational Methods in Engineering* 26 (2019), pp. 163–243. DOI: 10.1007/s11831-017-9232-5.

[70] J. Heyman. *Structural Analysis: A Historical Approach*. Cambridge University Press, 1998.

[71] A. Öchsner. *Classical Beam Theories of Structural Mechanics*. Springer International Publishing, Jan. 2021, pp. 1–186. DOI: 10.1007/978-3-030-76035-9.

[72] Dassault Systèmes SIMULIA. *Abaqus 2016 Analysis User's Guide*. Dassault Systèmes. Providence, RI, USA, 2015.

[73] G. Kirchhoff. "Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes." ger. *Journal für die reine und angewandte Mathematik* 56 (1859), pp. 285–313.

[74] A. E. H. Love. *A treatise on the mathematical theory of elasticity*. Fourth. Cambridge University Press, 1927.

[75] E. Reissner. "On One-Dimensional Finite-Strain Beam Theory". *Journal of Applied Mathematics and Physics (ZAMP)* 23 (1972).

[76] E. Reissner. "Reflections on the theory of elastic plates". *Applied Mechanics Review* 38.11 (1985), pp. 1453–1464. DOI: 10.1115/1.3143699.

[77] J. Simo. "A finite strain beam formulation. The three-dimensional dynamic problem. Part I". 49.1 (1985), pp. 55–70. DOI: https:/s/doi.org/10.1016/0045-7825(85)90050-7.

[78] M. A. Crisfield and G. Jelenic. "Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 455 (1983 1999), pp. 1125–1147. DOI: 10.1098/rspa.1999.0352.

[79] J. C. Simo and L. Vu-Quoc. *A three-dimensional finite-strain rod model. Part II: Computational aspects*. 1986.

[80] C. Meier, A. Popp, and W. A. Wall. "A locking-free finite element formulation and reduced models for geometrically exact Kirchhoff rods". *Computer Methods in Applied Mechanics and Engineering* 290 (2015), pp. 314–341. DOI: 10.1016/j.cma.2015.02.029.

[81] M. Schulz and M. Böl. "A finite element formulation for a geometrically exact Kirchhoff–Love beam based on constrained translation". *Computational Mechanics* 64 (4 Oct. 2019), pp. 1155–1175. DOI: 10.1007/s00466-019-01698-z.

[82] P. Wriggers. *Computational Contact Mechanics*. Berlin: Springer, 2006.

[83] A. Signorini. *Sopra alcune questioni di statica dei sistemi continui*. 1933.

[84] N.J. Carpenter, R.L. Taylor, and M.G. Katona. "Lagrange constraints for transient finite-element surface-contact". *International Journal for Numerical Methods in Engineering* 32 (1991), pp. 103–128.

[85] P. Wriggers and G. Zavarise. "On contact between three-dimensional beams undergoing large deflections". *Communications in Numerical Methods in Engineering* 13.6 (1997), pp. 429–438. DOI: 10.1002/(SICI)1099-0887(199706)13:6<429::AID-CNM70>3.0.CO;2-X.

[86] A. Sedmak. "Computational fracture mechanics: An overview from early efforts to recent achievements". *Fatigue & Fracture of Engineering Materials & Structures* 41.12 (2018), pp. 2438–2474. DOI: https://doi-org.tudelft.idm.oclc.org/10.1111/ffe.12912. eprint: https://onlinelibrary-wiley-com.tudelft.idm.oclc.org/doi/pdf/10.1111/ffe.12912.

[87] K. Park and G. H. Paulino. "Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces". *Applied Mechanics Reviews* 64.6 (Feb. 2013), p. 060802. DOI: 10.1115/1.4023110. eprint: https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article-pdf/64/6/060802/6073587/amr_64_6_060802.pdf.

[88] M. Elices, G. Guinea, J. Gómez, and J. Planas. "The cohesive zone model: advantages, limitations and challenges". *Engineering Fracture Mechanics* 69.2 (2002), pp. 137–163. DOI: https://doi.org/10.1016/S0013-7944(01)00083-2.

[89] Y. Liu, Z. Qu, and Y. Li. "Design Method of the Conformal Lattice Structures". *Advanced Engineering Materials* 24 (May 2022). DOI: 10.1002/adem.202101423.

[90] C. A. Meier. "Geometrically exact finite element formulations for slender beams and their contact interaction". PhD thesis. Technische Universität München, 2016.

[91] L. Noels and R. Radovitzky. "A general discontinuous Galerkin method for finite hyperelasticity. Formulation and numerical applications". *International Journal for Numerical Methods in Engineering* 68.1 (2006), pp. 64–97. DOI: 10.1002/nme.1699.

[92] G. Becker and L. Noels. "A fracture framework for Euler–Bernoulli beams based on a full discontinuous Galerkin formulation/extrinsic cohesive law combination". 85 (2011), pp. 1227–1251. DOI: 10.1002/nme.3008.

[93] G. Noh and K.-J. Bathe. "An explicit time integration scheme for the analysis of wave propagations". *Computers & Structures* 129 (2013), pp. 178–193. DOI: https://doi.org/10.1016/j.compstruc.2013.06.007.

[94] T. J. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. New York: Dover Publications, Inc, 2000.

[95] C. Meier, W. Wall, and A. Popp. "A unified approach for beam-to-beam contact". 315 (2017), pp. 972–1010. DOI: 10.1016/j.cma.2016.11.028.

[96] J. T. Sonandres. "A Computational Framework for Simulating Entanglement-Based Drone Countermeasures with Flexible Filaments Immersed in Viscous Flow". MA thesis. Massachusetts Institute of Technology, 2025.

[97] P. Wriggers. "Finite element algorithms for contact problems". *Archives of Computational Methods in Engineering* 2 (Dec. 1995), pp. 1–49. DOI: 10.1007/BF02736195.

[98] P. Wriggers, T. Vu Van, and E. Stein. "Finite element formulation of large deformation impact-contact problems with friction". *Computers & Structures* 37.3 (1990), pp. 319–331. DOI: https://doi.org/10.1016/0045-7949(90)90324-U.

[99] J. Case, L. Chilver, and C. T. Ross. "2 - Pin-jointed frames or trusses". *Strength of Materials and Structures (Fourth Edition)*. Fourth Edition. London: Butterworth-Heinemann, 1999, pp. 55–66. DOI: https://doi.org/10.1016/B978-034071920-6/50006-6.

[100] S. K. Kota, S. Kumar, S. Turteltaub, and B. Giovanardi. "Large-scale numerical modeling of fracture in truss architected materials" (2026 (submitted)).

[101] R. Zaharia and D. Dubina. "Stiffness of joints in bolted connected cold-formed steel trusses". *Journal of Constructional Steel Research* 62.3 (2006), pp. 240–249. DOI: https://doi.org/10.1016/j.jcsr.2005.07.002.

[102] G. Karypis and V. Kumar. *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*. http://www.cs.umn.edu/~metis. University of Minnesota, Minneapolis, MN, 2009.

[103] B. Giovanardi, S. Serebrinsky, and R. Radovitzky. "A fully-coupled computational framework for large-scale simulation of fluid-driven fracture propagation on parallel computers" (2020).

[104] A. Koshakji, G. Chomette, J. Turner, J. Jablonski, A. Haynes, D. Carlucci, B. Giovanardi, and R. A. Radovitzky. "A robust computational framework for simulating the dynamics of large assemblies of highly-flexible fibers immersed in viscous flow". *Journal of Computational Physics* 474 (2023), p. 111774. DOI: https://doi.org/10.1016/j.jcp.2022.111774.

[105] R. Fitzpatrick. *Classical Mechanics*. 2006. URL: https://farside.ph.utexas.edu/teaching/301/301 (visited on 09/23/2026).

[106] F. Cirak and M. West. "Decomposition contact response (DCR) for explicit finite element dynamics". *International Journal for Numerical Methods in Engineering* 64 (2005), pp. 1078–1110. DOI: 10.1002/nme.1400.

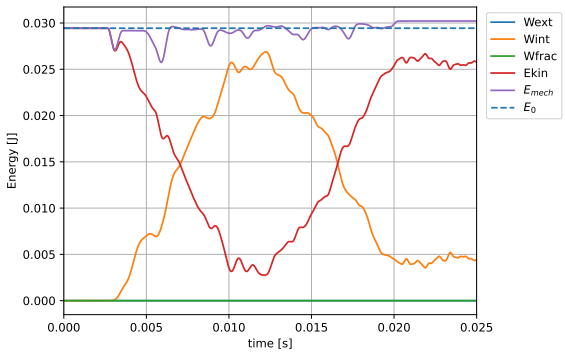
[107] P. Areias, J. Garcao, E. B. Pires, and J. I. Barbosa. "Exact corotational shell for finite strains and fracture". *Computational Mechanics* 48.4 (Apr. 2011), pp. 385–406. DOI: 10.1007/s00466-011-0588-3.

A

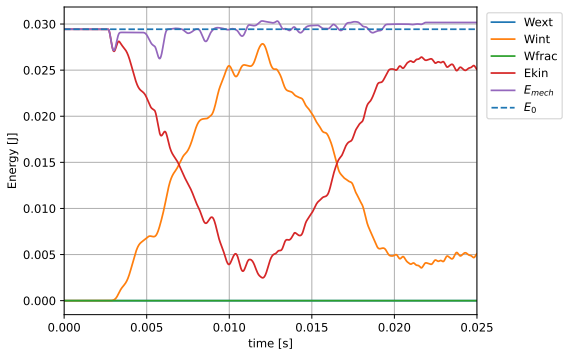
Additional figures

A.1. Additional figures for Subsection 4.3.2

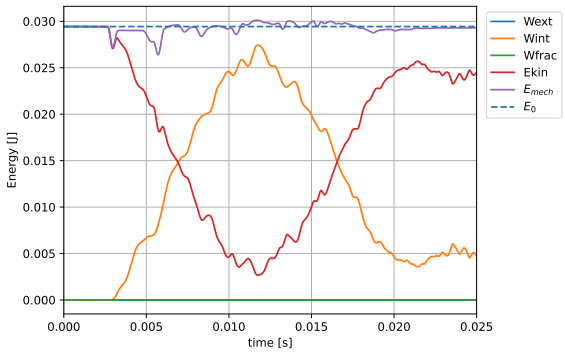
Subsection 4.3.2 only showed the energy plots for $R_{\text{contact}} = 2.5$ and $R_{\text{contact}} = 20.0$. Below, the energy plots for the other values of R_{contact} are also included.



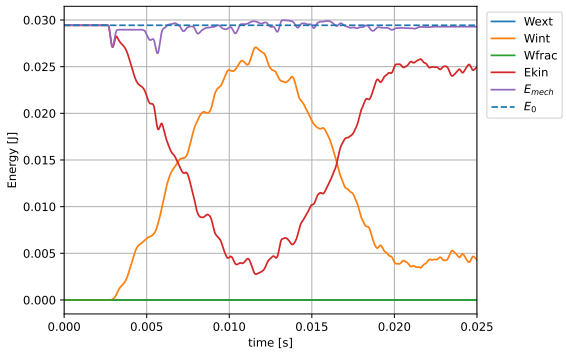
(a) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 5.0$.



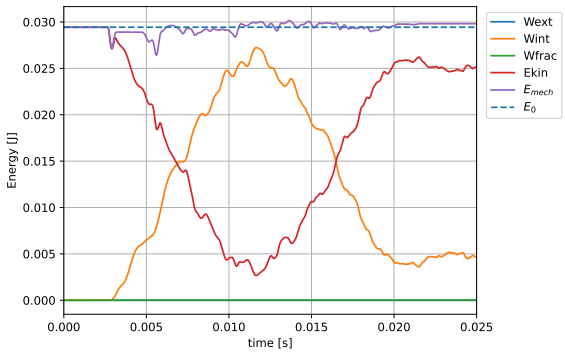
(b) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 7.5$.



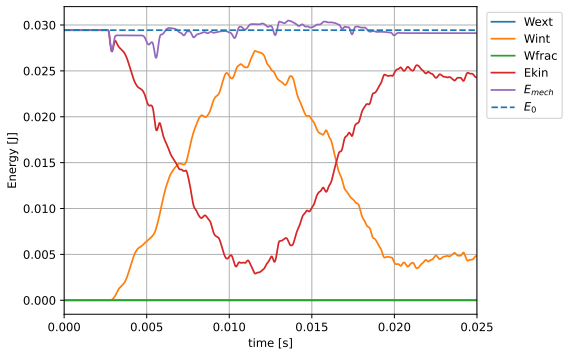
(c) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 10.0$. This figure is the same as Figure 4.17.



(d) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 12.5$.



(e) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 15.0$.

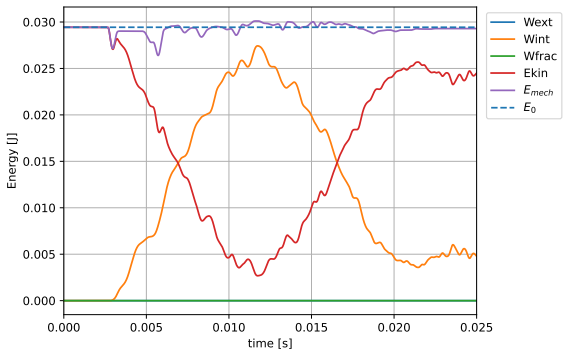


(f) The energy balance of the system from case III for $v_0 = -3.0$ m/s and $R_{\text{contact}} = 17.5$.

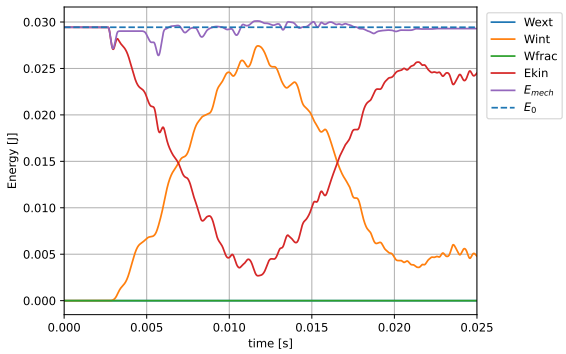
Figure A.1: The energy plots for the other values of R_{contact} from the contact penalty sensitivity study from Subsection 4.3.2.

A.2. Additional figures for Subsection 4.3.3

The energy plots for the smaller time step values used in Figure 4.22 have been included below.



(a) The energy balance of the system from case III for $v_0 = -3.0$ m/s, $R_{\text{contact}} = 10.0$ and $\Delta t = 0.5 \cdot 10^{-8}$ s.



(b) The energy balance of the system from case III for $v_0 = -3.0$ m/s, $R_{\text{contact}} = 10.0$ and $\Delta t = 1.0 \cdot 10^{-9}$ s.

Figure A.2: The energy plots for $\Delta t = 0.5 \cdot 10^{-8}$ s and $\Delta t = 1.0 \cdot 10^{-9}$ s from the convergence study from Subsection 4.3.3.

B

Template license

This thesis utilised a modified template by Daan Zwaneveld, which is licensed under CC BY-NC 4.0. To view a copy of this license, visit `https://creativecommons.org/licenses/by-nc/4.0/`.