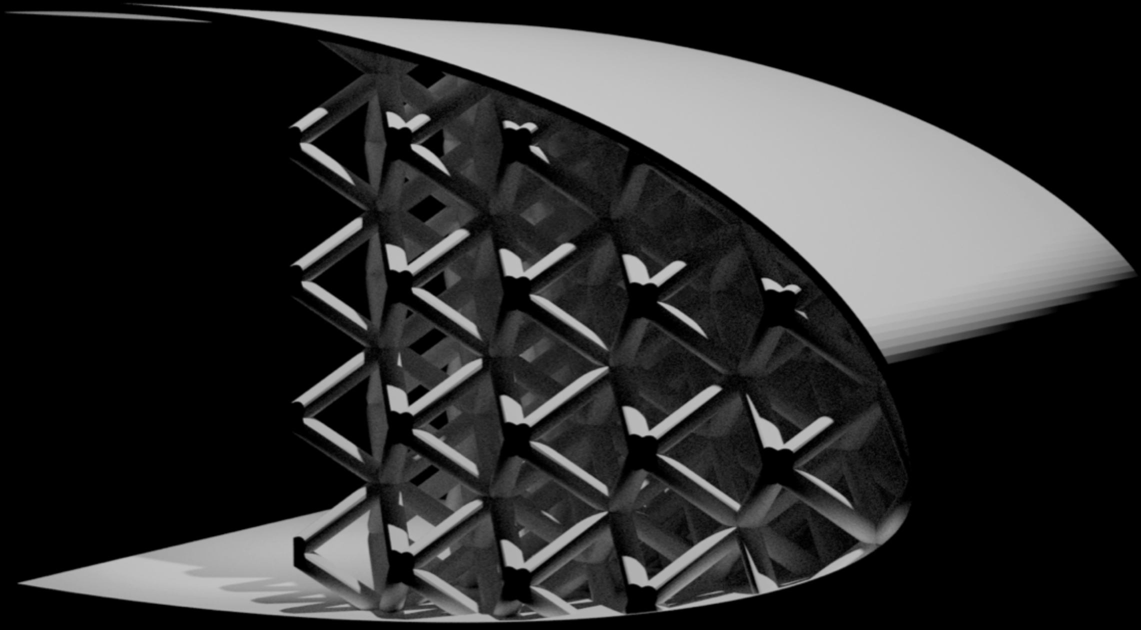
Hybrid Beam-3D Continuum Nonlinear Modeling of Architected Truss Metamaterials

Insights into Joint Geometry

AE5722: Thesis Aerospace Structures and Materials

Samuel Demian

Delft University of Technology



TU Delft

Hybrid Beam-3D Continuum Nonlinear Modeling of Architected Truss Metamaterials

Insights into Joint Geometry

by

Samuel Demian

to obtain the degree of Master of Science
at the Delft University of Technology Faculty of Aerospace Engineering,
to be defended publicly on Tuesday May 19, 2026 at 9:30 AM.

Student number:	5250935	
Project duration:	June 9th, 2025 – May 19th, 2026	
Thesis committee:	Dr. B. Giovanardi,	TU Delft, supervisor
	Dr. S. Kumar,	TU Delft, supervisor
	Dr. S. Turteltaub	TU Delft, chair
	Dr. K. Masania	TU Delft, external member

Cover: Airfoil Tip with Truss Energy Dissipating Metamaterial by S. Demian

Preface

Dear reader,

This thesis is the end of my journey from bachelor's student to master's graduate at TU Delft's Faculty of Aerospace Engineering. It is the result of almost a year of work, the product of many conversations and many frustrations, but also celebrations along the way. This long process was of course not the result of my work alone, but also of the support I got from many people. Of course, writing a preface without any name-drops, as Shawn would say, would be a crime. However, to avoid showing personal preferences, I am deliberately keeping it short on those.

First of all, I would like to thank my parents and family for their support and encouragement throughout my studies. I am grateful to have had the opportunity to pursue my degree with the support coming from all of you, support I needed to get this far.

Secondly, I want to express gratitude to my main supervisor, Bianca, for her guidance, encouragement and patience in helping me work through many issues I encountered along the way. I also want to give a big thank you to Bharad, for his help and support, especially in the later stages of the project when implementation issues were coming up and my hopes for good results were fading. It is definitely thanks to you two that my project reached this final form. I do not want to forget Sid, my secondary supervisor at the Faculty of Mechanical Engineering, for his help and support, especially in the early stages of the project when I was still trying to find my way through this new and exciting topic.

Finally, I want to thank all of my old friends, but also the new friends made along the way during my years in Delft, and abroad since the start of my studies for their encouragement, good memories shared, but also support. I believe that it is through this mutual support that most of us made it this far and found the path to do what excites us, challenges us, and, above all, brings us happiness.

Samuel Demian
Delft, May 2026

Executive Summary

Truss lattice metamaterials have attracted growing interest because their effective mechanical response can be tailored through architecture rather than through constituent material alone. This makes them promising candidates for lightweight structural design, energy absorption, and multifunctional aerospace applications. Within this broad field, low-relative-density lattices are especially attractive because they offer high stiffness- and strength-to-weight potential. However, they are also difficult to model reliably: as relative density decreases, the members become more slender, geometric instability becomes more important, and the response becomes increasingly sensitive to local geometric details such as the shape of the lattice joints.

Existing modeling approaches reflect a persistent trade-off. Pure beam and truss models are computationally efficient and scale well to large specimens, but they idealize the joint regions and therefore struggle to capture joint-sensitive mechanics. Fully resolved solid finite element models provide high geometric fidelity and can represent local stress concentrations, contact, and nonlinear material behavior, but they quickly become computationally prohibitive for low-density finite-sized lattice problems. The literature review in this thesis therefore identifies a methodological gap: there is a need for modeling strategies that retain local fidelity at the joints while remaining scalable enough for systematic studies of nonlinear lattice behavior.

To address this problem, this thesis develops a nonlinear-compatible hybrid beam–3D continuum modeling approach for truss lattice metamaterials. In the proposed framework, the joint regions are discretized with three-dimensional continuum elements, while the connecting struts are represented with beam elements. The coupling between both submodels is formulated as an interface-equilibrium problem, in which beam-end kinematics are mapped to solid-face motion and reaction resultants are exchanged iteratively between the subdomains. This partitioned formulation preserves a clear separation between beam and solid solvers and is designed to remain compatible with geometric nonlinearity and elastoplastic extensions.

A second contribution of the thesis is algorithmic. Because the hybrid formulation leads to a nonlinear interface problem, dedicated iterative solution procedures are developed and discussed. These include fixed-point methods, Anderson acceleration, preconditioned residual descent, full-Jacobian Newton strategies, and Jacobian-free Newton–Krylov methods. Their purpose is not only to achieve convergence, but also to preserve the modular and parallel character of the hybrid modeling framework. In that sense, the thesis treats the hybrid approach not merely as a discretization choice, but as a coupled computational architecture.

The results show that the hybrid beam–3D continuum formulation provides a useful intermediate modeling strategy between pure beam models and fully resolved solid finite element models. Compared with pure beam discretizations, the hybrid model retains an explicit representation of the joint regions and is therefore better suited to studying joint-geometry effects. Compared with fully solid models, it substantially reduces the number of three-dimensional continuum degrees of freedom by concentrating solid discretization in the joint regions. Across the investigated accuracy cases, the hybrid formulation reduces the degree-of-freedom count by approximately 54–94% relative to the corresponding solid element references, while the available simulation logs show peak-memory reductions of approximately 95–99%. The scaling study further indicates that the cost of the hybrid method is governed mainly by the repeated local solid-joint solves, the chosen joint resolution, and the number of residual evaluations required by the staggered nonlinear coupling, while parallel execution can reduce wall time when enough independent subproblems are available.

The accuracy results show that the hybrid model is most valuable in regimes where pure beam idealization removes mechanically important joint geometry. In slender stretching-dominated octet lattices, the beam-only models can reproduce the initial effective stiffness reasonably well, but they delay the apparent onset of nonlinear softening and overpredict the post-buckling stress. In these cases, the hybrid model remains closer to the solid element reference in the nonlinear response because it retains the finite joint geometry. In bending-dominated pyramidal lattices, the hybrid model recovers a substantial part of the stiffness and energy absorption missing from the beam-only models, but its absolute agreement with the solid reference remains less controlled and can include overprediction. The hybrid formulation should therefore not be interpreted as a uniformly conservative or uniformly exact replacement for full solid finite element modeling. Its main contribution is instead to provide a nonlinear-compatible, joint-sensitive, and computationally reduced framework for comparative lattice studies, while keeping the remaining case-dependent modeling error explicit. Within these limits, the proposed framework provides a practical basis for further investigation of joint-geometry design in architected truss metamaterials.

Contents

List of Figures

List of Tables

Symbols

Symbol	Definition	Unit
Latin symbols		
A	Cross-sectional area	[-]
C_E	Stiffness scaling coefficient	[-]
C_σ	Strength scaling coefficient	[-]
d	Strut diameter	[-]
E	Young's modulus	[-]
E_s	Young's modulus of the parent solid material	[-]
E^*	Effective Young's modulus of the lattice	[-]
E_0^*	Initial effective Young's modulus of the lattice	[-]
$\mathbf{f}_{\Gamma_i}^{(s)}$	Solid-side interface resultant at interface Γ_i	[-]
$\mathbf{f}_{\Gamma_i}^{(b)}$	Beam-side interface resultant at interface Γ_i	[-]
$\mathbf{f}_{\text{int}}$	Internal force vector	[-]
$\mathbf{f}_{\text{ext}}$	External force vector	[-]
$\mathbf{F}_{\Gamma_i}^{(s)}$	Solid-side resultant force at interface Γ_i	[-]
h	Finite-difference perturbation size	[-]
I	Second moment of area	[-]
$\mathbf{J}$	Jacobian matrix of the interface residual	[-]
K	Stiffness matrix or tangent stiffness matrix	[-]
$\widetilde{\mathbf{J}}$	Sparse approximate interface Jacobian	[-]
L	Unit-cell side length	[-]
L_{UC}	Reference unit-cell length	[-]
l	Characteristic strut length	[-]
l_{mod}	Modified cross-section length near the joint	[-]
l_{trans}	Transition length from modified section to circular strut section	[-]
M	Moment or moment resultant	[-]
M_{solid}	Memory required by a fully solid model	[GB]
M_{hyb}	Memory required by a hybrid model	[GB]
n	Size of the global interface vector	[-]
n_E	Stiffness scaling exponent	[-]
n_σ	Strength scaling exponent	[-]
N_x, N_y, N_z	Number of repeated unit cells in the x, y, and z directions	[-]
N_B	Number of beam submodels	[-]
N_J	Number of solid joint submodels	[-]
N_Γ	Number of beam–solid coupling interfaces	[-]
N_{dof}	Number of degrees of freedom	[-]
$\mathbf{P}$	Preconditioner	[-]
$\mathbf{q}$	Global stacked interface unknown vector	[-]
$\mathbf{q}_{\Gamma_i}$	Six-component interface state at interface Γ_i	[-]
$\mathbf{q}_{\Gamma_i}^{(s)}$	Solid-side rigid-face interface motion at interface Γ_i	[-]
$\mathbf{q}_{\Gamma_i}^{(b)}$	Beam-side interface motion at interface Γ_i	[-]
$\Delta\mathbf{q}$	Interface-state correction vector	[-]
r	Strut radius	[-]

Symbol	Definition	Unit
r/l	Strut radius-to-length ratio	[-]
$\mathbf{r}_{\Gamma_i}$	Interface residual at interface Γ_i	[-]
$\mathbf{R}$	Global nonlinear residual vector	[-]
$\mathbf{R}_e$	Element corotational transformation matrix	[-]
T_{solid}	Wall time of a fully solid analysis	[s]
T_{hyb}	Wall time of a hybrid analysis	[s]
$\mathbf{t}_{\Gamma_i}$	Translational part of an interface motion	[-]
$\mathbf{u}^{(s)}$	Solid displacement field	[-]
V_{circ}	Volume of the circular-joint geometry	[-]
V_{sq}	Volume of the square-joint geometry	[-]
W	Energy absorbed per unit volume	[-]
w	Energy absorbed per unit mass	[-]
$\mathbf{x}$	Position vector	[-]
$\mathbf{x}_{c,i}$	Centroid position of interface face Γ_i	[-]
Greek symbols		
α	Load-step or line-search step length	[-]
Γ_i	Beam–solid coupling interface i	[-]
ε	Macroscopic strain	[-]
λ	Load-ramp parameter	[-]
ν	Poisson's ratio	[-]
ρ	Density	[-]
ρ_s	Density of the parent solid material	[-]
ρ^*	Apparent density of the lattice material	[-]
$\bar{\rho}$	Relative density of the lattice, ρ^*/ρ_s	[-]
σ	Macroscopic stress	[-]
σ_s	Strength or stress scale of the parent solid material	[-]
σ^*	Effective strength or stress measure of the lattice	[-]
$\boldsymbol{\theta}_{\Gamma_i}$	Rotational part of an interface motion	[-]
Ω_s	Solid-joint domain	[-]
Ω_b	Beam-strut domain	[-]
ω	Relaxation factor in fixed-point or Aitken-type iteration	[-]

1

Introduction

Architected truss metamaterials derive their mechanical response not only from the properties of the parent material, but also from the geometry of a repeated structural network. This ability to tune stiffness, strength, anisotropy, and collapse behavior through architecture has made truss metamaterials increasingly attractive for lightweight structural design, energy absorption, and multifunctional aerospace components. In particular, low-relative-density trusses are of strong interest because they offer access to combinations of mass efficiency and mechanical performance that are difficult to achieve with conventional monolithic materials. At the same time, this regime is also one of the most difficult to analyze reliably, since deformation becomes increasingly sensitive to member slenderness, instability, imperfections, and local geometric detail.

In idealized beam descriptions, joints are treated as zero-volume points, which is often sufficient for first-order predictions in slender architectures. However, real truss metamaterials contain finite joint regions whose geometry can alter stiffness, stress redistribution, and the onset of nonlinear deformation. The local geometry of the joints cannot always be neglected. This issue becomes especially relevant when studying non-slender geometries, modified joint shapes, or nonlinear response under large deformation and plasticity. Literature indicates both that joint geometry can materially influence truss metamaterial behavior and that current modeling strategies struggle to capture these effects while remaining computationally scalable.

This thesis addresses this gap through the development of a nonlinear-compatible hybrid beam–3D continuum modeling framework for truss metamaterials. The underlying idea is to retain three-dimensional continuum fidelity where it matters most, namely in the joint regions, while representing the slender struts with beam elements in order to preserve tractability for larger models. In contrast to reduced-order linear strategies based on static condensation, the present approach is formulated to remain compatible with geometric nonlinearity and, ultimately, with elastoplastic response. Within the current formulation, the method is posed as a partitioned interface-equilibrium problem in which solid-node and beam-strut submodels are coupled iteratively through interface displacements, rotations, forces, and moments.

The scientific objective of this thesis is therefore twofold. First, it seeks to establish whether a hybrid beam–3D continuum discretization can provide a mechanically credible and computationally attractive alternative to purely beam-based or fully solid finite element models for truss metamaterials. Second, it seeks to use that framework to enable systematic investigation of how joint geometry influences the nonlinear response of low-relative-density truss metamaterials. Special attention is given to the role of modified joint cross-sections, to the interaction between joint effects and architecture type, and to the question of whether improved joint modeling can reveal trends that are inaccessible to simpler discretizations.

To support these objectives, the thesis is structured as follows. Chapter 2 reviews the literature on truss metamaterials, low-relative-density behavior, joint geometry effects, and modeling trade-offs.

Chapter 3 introduces the governing equations, geometric parameter definitions, periodic solid-element reference models, and the hybrid beam–3D continuum formulation. Chapter 4 presents the iterative coupling algorithms used to solve the interface equilibrium problem, and discusses the scalability of the method; including small-scale scaling experiments. Chapter 5 validates the solid, beam, and hybrid models through mesh convergence studies and comparisons against reference solid-element simulations, followed by an accuracy sweep across different geometries comparing the three modeling approaches. This chapter contains the main results motivating the contribution of the thesis. Finally, Chapter 6 summarizes the main conclusions, and chapter 7 gives recommendations for future work.

The contribution of this thesis is therefore the development and assessment of a nonlinear-compatible, joint-sensitive modeling framework for architected truss metamaterials. The framework is evaluated as an intermediate approach between pure beam and fully solid finite element models. Its usefulness is judged by its ability to reproduce solid-reference trends, preserve joint-geometry effects, and reduce computational cost.

2

Literature Review

Although the term "metamaterial" became prominent through the study of materials with engineered electromagnetic properties, it has since been adopted more broadly to describe materials whose unusual properties arise from internal architecture rather than from composition alone. In mechanics, this includes negative Poisson's ratio, or auxetic, materials and, more recently, architected cellular and lattice materials. A foundational early example is the work of Lakes [1], which demonstrated negative Poisson's ratio behavior in a foam structure. The broader field of architected mechanical metamaterials gained substantial momentum in the 2000s and 2010s, driven by advances in microfabrication and additive manufacturing that made increasingly complex cellular and lattice architectures experimentally realizable [2–7].

This chapter reviews the literature on architected lattice metamaterials and modeling approaches in order to position the present work. Particular emphasis is placed on the mechanical behavior of low-relative-density truss lattices, the role of joint geometry, and the limitations of current modeling methods when these effects must be studied systematically.

The chapter is organized as follows. Section 2.1 introduces lattice metamaterials as architected mechanical systems and reviews the main architecture–property relationships used to describe them. Section 2.2 then explains why the low-relative-density regime is especially attractive, but also especially difficult to analyze and predict. Section 2.3 discusses why joint geometry can act as an important secondary design variable beyond topology and relative density alone. Section 2.4 reviews the main modeling approaches used in the literature, including beam and truss idealizations, full solid finite element models, and hybrid beam–3D continuum or reduced-order joint-resolving strategies. Finally, section 2.5 identifies the research gap addressed in this thesis and positions the contribution of the present work within that context.

2.1. Lattice Metamaterials: Architecture and Mechanical Behavior

Architected lattice metamaterials are cellular solids in which the macroscopic mechanical response is governed primarily by an intentionally designed internal geometry rather than by the parent material alone [2–4, 6, 8]. Unlike stochastic foams, such materials are typically based on a repeatable unit cell and can therefore be tuned in a controlled way through topology, member dimensions, and spatial arrangement [6, 8, 9].

This section first defines lattice metamaterials and situates them among related architected cellular materials in subsection 2.1.1. It then introduces relative density as a central design variable in subsection 2.1.2, before reviewing the architecture–property scaling relations commonly used in the literature in subsection 2.1.3. Finally, subsection 2.1.4 discusses the distinction between bending-dominated and stretching-dominated topologies, which is essential for understanding why some lattice families are especially promising for lightweight structural applications.

2.1.1. Definition and Classification of Lattice Metamaterials

Architected lattice metamaterials, also referred to in the literature as architected cellular materials, are solids whose effective properties arise primarily from geometry [4, 6, 7]. In contrast to conventional bulk solids, their stiffness, strength, anisotropy, and deformation behavior are not determined only by the parent material, but also by the topology and dimensions of a repeated structural network [2–4]. Typical lattice families include body-centered cubic (BCC), face-centered cubic (FCC), octet-truss, Kelvin, and pyramidal topologies, among many others [6, 8, 10, 11].

Representative examples of common lattice families are shown in Figure 2.1, together with their broad classification into bending-dominated and stretching-dominated architectures. This classification is based on whether the dominant load-transfer mechanism is member bending or axial stretching and compression [10, 11].

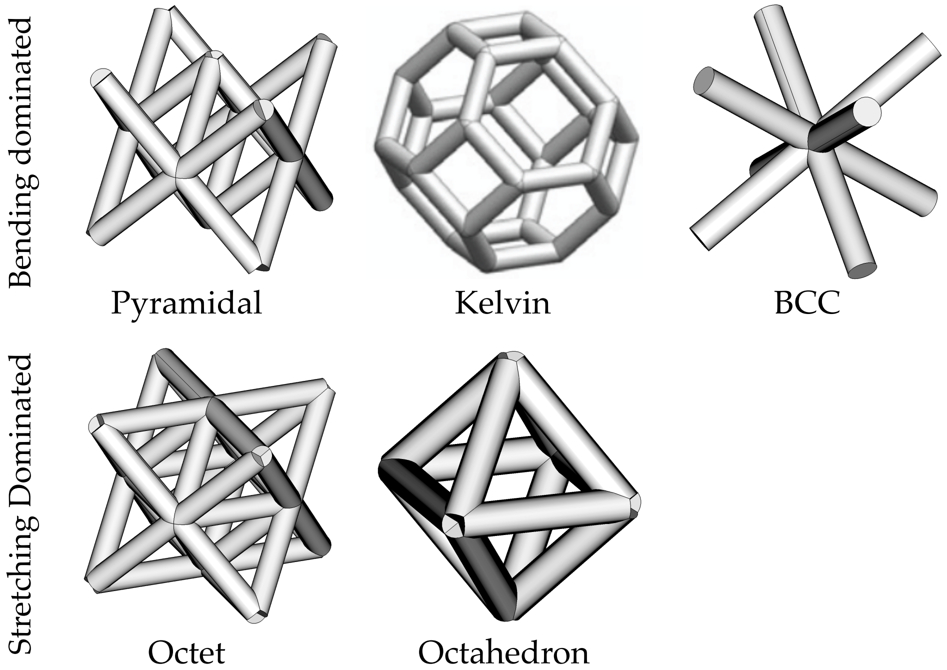


Figure 2.1: Representative lattice topologies used throughout the literature. The figure distinguishes bending-dominated architectures, namely pyramidal, Kelvin, and BCC, from stretching-dominated architectures, namely octet and octahedron. The classification follows the common distinction between bending-dominated and stretching-dominated cellular architectures [8, 10, 11].

These lattices may be arranged periodically, graded spatially, or otherwise modified in order to tailor the resulting mechanical response [2, 3, 6, 9]. This design freedom is one of the main reasons for the strong research interest in these materials, especially in applications where weight efficiency is critical. In such applications, architected lattices offer the possibility to design specific stiffness, specific strength, anisotropy, and nonlinear collapse behavior more directly than would be possible with a homogeneous material alone [2–6].

The appeal of such architectures lies in the fact that they can occupy regions of mechanical property space that are difficult or impossible to achieve using conventional monolithic materials. Portela et al. [12] showed that three-dimensional truss lattice architectures can exhibit combinations of mechanical properties that are difficult to obtain in homogeneous solids, while Zheng et al. [13] and Meza et al. [14] demonstrated that carefully designed micro- and nanolattice architectures can achieve exceptional stiffness- and strength-to-weight performance at extremely low density. In particular, the combination of low weight and topology-controlled load paths makes lattice metamaterials attractive for stiffness-critical structures, energy absorbers, and multifunctional components [2, 3, 5, 15, 16].

2.1.2. Relative Density as a Governing Design Parameter

A convenient macroscopic parameter linking lattice architecture to performance is the relative density given by Equation 2.1,

$$\bar{\rho} = \frac{\rho^*}{\rho_s}, \tag{2.1}$$

where ρ^* is the apparent density of the lattice material, defined as mass divided by envelope volume, and ρ_s is the density of the parent solid. Gibson and Ashby [8] established relative density as a central descriptor in cellular solids because it simultaneously captures weight efficiency and provides the natural independent variable for scaling relations of effective stiffness and strength.

Because relative density depends directly on the amount of solid material distributed within the lattice envelope, it is also an intuitive way to compare different topologies at approximately equal mass [8, 10, 11]. This makes it especially useful in lightweight design, where the central question is not only absolute performance, but performance at a given structural weight. For this reason, much of the mechanical lattice literature is organized around the effect of changing $\bar{\rho}$, either within a given topology or when comparing different topologies [4, 6, 8].

Relative density is also the parameter that most clearly exposes the design trade-off at the heart of lattice metamaterials. Decreasing $\bar{\rho}$ improves weight efficiency, but it also tends to increase member slenderness and reduce the margin to instability [8, 10, 17]. The resulting behavior is therefore not captured by mass alone: two lattices with the same relative density may differ substantially in stiffness, failure mode, and energy absorption depending on the underlying architecture and on the geometry of the strut–joint transitions [10–12, 18, 19].

2.1.3. Architecture–Property Scaling Laws

A large fraction of the lattice metamaterials literature rationalizes effective properties through power-law relationships in relative density [8, 10, 11], commonly written as

$$\frac{E^*}{E_s} = C_E \bar{\rho}^{\,n_E}, \qquad \frac{\sigma^*}{\sigma_s} = C_\sigma \bar{\rho}^{\,n_\sigma},$$

where E^* and σ^* denote effective elastic and strength measures of the lattice, E_s and σ_s are the corresponding properties of the parent material, and C_E, n_E, C_σ, and n_σ are constants that depend on the lattice architecture.

Within the classical cellular-solids framework, Gibson and Ashby [8] showed that the exponent n_E is strongly linked to the dominant deformation mechanism. Stretching-dominated lattices typically show stiffness scaling closer to $n_E \approx 1$, whereas bending-dominated lattices more often exhibit $n_E \approx 2$, with analogous distinctions commonly reported for strength scaling. Deshpande et al. [10] further formalized the importance of this distinction by relating deformation mode to structural connectivity and to the resulting efficiency of the architecture. Hutchinson and Fleck [11] extended this line of reasoning for periodic trusses by showing that structural performance depends strongly on topology and load-transfer mechanism, thereby reinforcing the central role of architecture in determining stiffness and strength at low weight.

At the same time, these scaling laws are idealizations. Glaesener et al. [17] demonstrated that geometric imperfections can significantly alter the mechanical response of periodic trusses, while Portela et al. [12] noted that simple analytical and computational descriptions lose predictive power once joint effects and non-slender geometries become important. As a consequence, topology and relative density remain the dominant first-order descriptors, but they are not always sufficient to explain the nonlinear response of real lattice specimens.

This observation is important for the present thesis because it motivates moving beyond architecture-level descriptors alone. If the response of slender or low-density lattices deviates from ideal scaling due to imperfections, instability, or local geometric effects, then modeling approaches able to capture those effects become valuable even when the global architecture remains unchanged [12, 17].

2.1.4. Deformation Regimes: Bending-dominated and Stretching-dominated Topologies

A central distinction in the mechanics of lattice metamaterials is whether global deformation is dominated by member bending and rotation or by axial stretching and compression [8, 10, 11]. This distinction is largely topological. Networks that admit mechanism-like motions can accommodate macroscopic strain with limited axial member strain, leading to bending-dominated behavior. By contrast, networks that suppress such mechanisms enforce axial load transfer more directly and tend to behave in a stretching-dominated manner [10, 11, 20].

Deshpande et al. [10] showed that this distinction can be understood through connectivity arguments and that stretching-dominated cellular architectures are generally more weight-efficient for structural applications than bending-dominated ones. This is a central reason why stretching-dominated truss topologies remain attractive for lightweight engineering applications.

This classification is important because it directly influences the architecture–property scaling introduced in subsection 2.1.3. Stretching-dominated lattices are often favored for lightweight load-bearing applications because they can retain stiffness more effectively as relative density decreases [10, 11]. However, this apparent advantage also comes with increased sensitivity to slenderness, instability, and local geometric effects once the structure enters nonlinear regimes [17, 21]. This makes the deformation mode distinction central not only for understanding performance, but also for understanding where idealized modeling approaches may become insufficient.

This distinction between bending-dominated and stretching-dominated behavior is also supported by later work on periodic trusses. Hutchinson and Fleck [11] showed that the structural performance of periodic truss architectures depends strongly on topology and on the extent to which loads are carried through axial member action rather than bending. This further supports the use of deformation regime as a central organizing concept in the mechanics of truss lattice metamaterials.

For the argument developed in this thesis, the stretching-dominated regime is especially relevant. It is the regime in which topology offers very attractive stiffness-to-weight performance, but it is also the regime in which localized buckling and geometric imperfections can quickly become decisive [10, 11, 17]. This combination makes it a natural target for improved modeling strategies.

2.2. Why Low-relative-density Metamaterial Behavior is Especially Interesting

For many attractive engineering applications of lattice metamaterials, the regime of greatest interest is low relative density. It is in this regime that the greatest mass savings can be achieved, but it is also where the mechanical response becomes especially sensitive to instability, imperfections, and local geometric details [2–4, 8, 17].

This section explains why low relative density is attractive from a structural design perspective in subsection 2.2.1, and why it is simultaneously difficult to analyze and predict in subsection 2.2.2. Together, these arguments motivate the need to go beyond idealized first-order descriptors and consider more carefully the local mechanisms that govern nonlinear response.

2.2.1. Why Low Relative Density is Attractive

For high-performance applications, and especially in aerospace engineering, the most compelling regime for lattice materials is often the low-relative-density regime [2–4, 15]. Reducing $\bar{\rho}$ can lead to major mass savings for a given stiffness target, or to improved energy absorption per unit mass in applications involving controlled collapse [3, 8, 21, 22]. Because the mechanical response of a lattice is architecture-dependent, the possibility of maintaining useful stiffness, strength, or crashworthiness at low density is one of the main attractions of these materials [5, 10, 11, 13, 14].

A useful nonlinear performance measure in this context is the energy absorption per unit volume or per unit mass given by Equation 2.2,

$$W(\varepsilon) = \int_0^{\varepsilon} \sigma(\tilde{\varepsilon})\, d\tilde{\varepsilon}, \qquad w(\varepsilon) = \frac{W(\varepsilon)}{\rho^*}, \tag{2.2}$$

where $\sigma(\varepsilon)$ is the macroscopic stress–strain response. This metric makes clear that both the shape of the stress–strain curve and the density contribute to the usefulness of a lattice in crashworthiness or energy absorption applications [21, 22].

Yin et al. [22] reviewed the use of lattice structures for energy absorption and showed that lightweight architected lattices constitute an important design class for crashworthiness applications. Sun et al. [21] further demonstrated that topology can strongly affect the deformation and energy absorption characteristics of additively manufactured lattice structures. Beyond this application-driven perspective, Zheng et al. [13] showed that ultralight microarchitected materials can retain remarkable stiffness efficiency at very low density, while Meza et al. [14] demonstrated that three-dimensional nanolattice architectures can combine low weight with high strength, recoverability, and substantial energy absorption. In that sense, the low-density regime is attractive not only because it reduces mass, but because it opens access to a mechanical property space that is difficult to realize using conventional bulk materials [4–6, 23].

This broader motivation is illustrated in Figure 2.2, adapted from Bauer et al. [23]. The left panel is a compressive strength–density material-property chart comparing nano-, micro-, and macrolattices with stochastic nanoporous foams and commercial bulk materials. In this representation, diagonal lines correspond to constant specific strength, so data lying further toward the upper left indicate materials combining low density with high strength. Bauer et al. [23] uses this figure to show that many architected materials, especially at small length scale, extend into the low-density/high-strength "white space" that is inaccessible to conventional bulk materials.

The right panel replots the data in terms of normalized strength, $\sigma_{\mathrm{eff}}/E_s$, as a function of relative density, $\bar{\rho}$. This removes much of the trivial influence of the constituent material stiffness and makes the role of architecture clearer. Bauer et al. [23] shows that in this representation the data can be interpreted relative to classical scaling laws associated with stretching-, bending-, and buckling-governed behavior. A central conclusion is that, at very low relative density, the influence of architecture becomes increasingly important: rigid topologies outperform non-rigid ones, and the performance gap between stretching-dominated and bending-dominated architectures becomes much more significant.

The literature on energy-absorbing and ultralight lattice structures therefore suggests that low-density truss metamaterials are interesting not merely as lighter versions of dense cellular materials, but as a distinct design class with tunable collapse behavior and unusual specific mechanical performance [5, 13, 14, 21–23]. In particular, Figure 2.2 supports the view that the low-relative-density regime is attractive not only because of mass reduction, but because it amplifies the mechanical consequences of topology and architecture.

2.2.2. Why Low Relative Density is Difficult

The same geometric features that make low-relative-density lattices attractive also make them difficult to model and predict. As $\bar{\rho}$ decreases, the struts become more slender and the governing response mechanisms shift away from simple material yield and toward a coupled combination of geometric nonlinearity, buckling, localization, and post-yield collapse [8, 10, 17, 21]. In this regime, relatively small geometric imperfections or local compliance changes can strongly influence where instability begins and how deformation propagates [12, 17, 18].

This difficulty is especially relevant for stretching-dominated truss lattices. Although their linear scaling behavior is often favorable, their nonlinear response can be governed by localized buckling or failure events that depend on specimen size, boundary conditions, and local geometry [10, 11, 17, 21]. As a result, idealized models that capture only topology and strut slenderness may not remain sufficiently predictive once the structure enters the strongly nonlinear regime [12, 21].

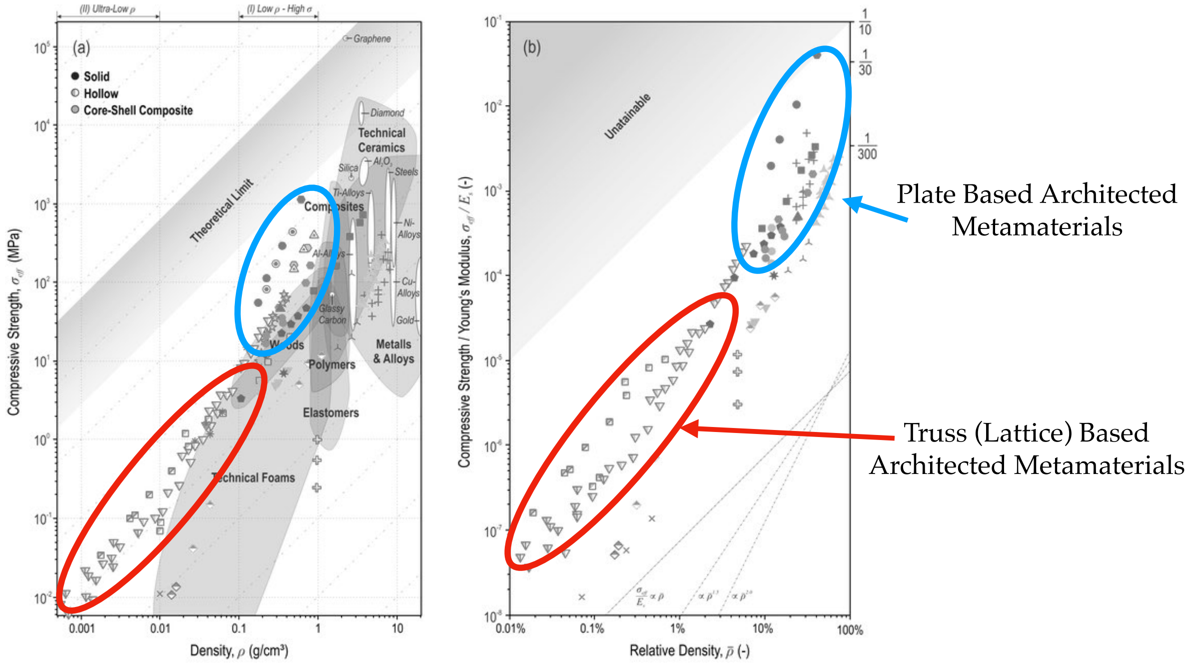


Figure 2.2: Low-density architected materials in mechanical property space, adapted from Bauer et al. [23]. Panel (a) shows compressive strength as a function of density for nano-, micro-, and macrolattices, stochastic nanoporous foams, and commercial bulk materials; the diagonal guidelines indicate constant specific strength, and the figure highlights the low-density/high-strength “white space” reached by some architected materials. Panel (b) shows normalized compressive strength, σ_{eff}/E_s, as a function of relative density, $\bar{\rho}$, allowing the data to be interpreted relative to classical stretching-, bending-, and buckling-type scaling trends. Together, the two panels show both that architected materials can access property regimes unavailable to conventional bulk solids and that architecture becomes increasingly decisive as relative density decreases.

Sun et al. [21] reported architecture-dependent crushing and energy absorption behavior in additively manufactured lattices, while Glaesener et al. [17] showed that geometric imperfections can have a marked influence on the response of both 2D and 3D periodic trusses. The low-relative-density regime is therefore not only mechanically attractive, but also particularly demanding from both an experimental and computational perspective.

This increasing sensitivity to local geometry is one reason why the low-relative-density regime forms a natural motivation for the present thesis. It suggests that effects originating at the joints may become disproportionately important, even when topology and relative density remain the dominant first-order descriptors [12, 18, 19].

2.3. Joint Geometry as a Secondary but Potentially Decisive Design Variable

Although topology and relative density are the dominant first-order descriptors of lattice metamaterials, they do not fully determine the nonlinear response of real structures [8, 10–12, 17]. In idealized truss theory and beam-based lattice modeling, a joint is often treated as a mathematical point that transfers force without additional compliance or stress concentration [10–12]. In real lattice structures, however, joints are finite-volume geometric regions whose shape can influence stiffness, stress redistribution, and the initiation of instability or plasticity [12, 18, 19, 24].

This section first contrasts idealized and real joint representations in subsection 2.3.1. It then discusses the physical mechanisms by which joint geometry can influence lattice response in subsection 2.3.2. Section 2.3.3 summarizes the relevant literature on this topic, and subsection 2.3.4 identifies the unresolved questions that most directly motivate the present work.

2.3.1. Idealized Joints versus Real Joint Regions

In analytical truss theory and in many beam-based lattice models, joints are idealized as zero-volume points connecting one-dimensional members [10–12, 25]. This abstraction is highly useful, and often sufficient for first-order predictions of global stiffness or topology-dependent behavior. However, real manufactured lattices do not contain such ideal joints. Instead, the connection between struts occupies a finite region of material whose geometry depends on both design intent and manufacturing constraints [5, 6, 9].

In additively manufactured lattices in particular, the joint region can deviate from nominal CAD (computer-aided design) geometry through over-thickening, blending, rounding, roughness, or local defects [6, 9]. These deviations can alter not only the local stress field, but also the local bending stiffness and effective rotational compliance of the connection [12, 18, 24]. As a result, the joint is not simply a geometric detail; it may actively influence the way load is redistributed between neighboring struts.

Portela et al. [12] introduced a simple beam-and-rotational-spring analogy to isolate the mechanical role of the joint in the most reduced possible setting. In their construction, Euler–Bernoulli beam elements of radius r, length l, and modulus E are connected through rotational springs of stiffness k_θ, so that the effect of the joint can be varied independently of the strut slenderness ratio r/l. This allows the nodal contribution to stiffness to be studied without yet committing to a full three-dimensional lattice model.

The conceptual role of local joint compliance is illustrated in Figure 2.3, adapted directly from Portela et al. [12]. The figure compares two representative cases: a rigid geometry in panel (a), and a non-rigid geometry in panel (b). In the rigid case, varying the rotational spring stiffness k_θ produces only small changes in the reduced structural stiffness, indicating that joint compliance has a limited influence on the overall response. In the non-rigid case, by contrast, the reduced stiffness depends strongly on k_θ, and in the limit of vanishing rotational stiffness the structure activates a zero-energy mechanism and loses stiffness entirely. In this way, Portela et al. [12] show that joint stiffness can be interpreted as a measure of nodal influence, and that this influence is expected to be much more pronounced in non-rigid than in rigid architectures.

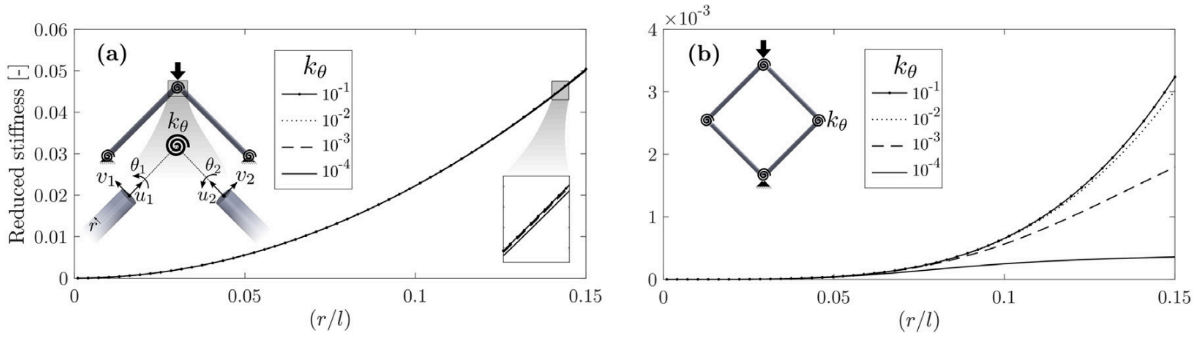


Figure 2.3: Effective structural stiffness of simple beam-theory models with rotational springs at the joints, adapted from Fig. 1 of Portela et al. [12]. Panel (a) shows a rigid geometry, for which varying the joint rotational stiffness k_θ has only a minor effect on the reduced stiffness. Panel (b) shows a non-rigid geometry, for which the reduced stiffness depends strongly on k_θ, because soft joints activate a zero-energy mechanism. The figure therefore illustrates a central point of this thesis: the mechanical importance of finite joint behavior depends strongly on the rigidity class of the underlying architecture.

As shown in Figure 2.3, the relevance of finite joint geometry is not uniform across lattice types. Rather, the Portela construction suggests that joint effects are expected to be weak in rigid architectures but potentially decisive in non-rigid architectures, which provides a useful conceptual foundation for the more detailed lattice studies discussed in the following subsections.

2.3.2. Mechanisms by which Joint Geometry Affects Response

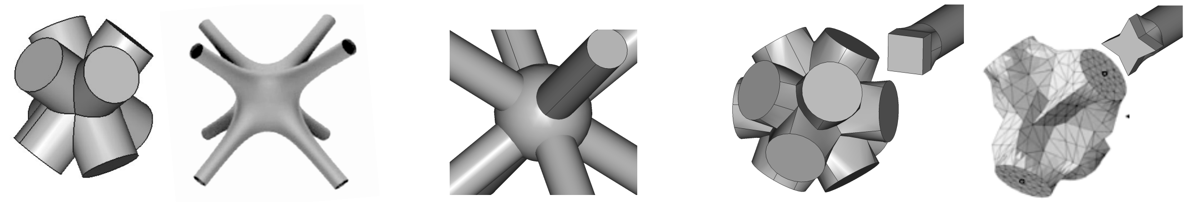
There are several mechanisms by which joint geometry may affect the mechanical response of a truss lattice. First, the local transition shape between strut and joint influences stress concentration and therefore the initiation of yielding or damage. Second, the amount of material concentrated near the joint modifies local bending and torsional stiffness, which can alter the effective compliance of the connection. Third, at fixed relative density, any increase in material allocated to the joints is accompanied by a corresponding redistribution of material away from the struts, which may reduce strut buckling resistance even if local stress concentrations are reduced [12, 18, 19, 24].

This trade-off is particularly important in low-relative-density lattices. In that regime, the nonlinear response can be controlled by the competition between local yielding near the joints and instability in the slender struts [17, 18, 21]. Joint geometry can therefore affect not only strength or initial stiffness, but also collapse mode selection, deformation localization, and energy absorption [12, 18, 19, 24]. Portela et al. [12] demonstrated that changes in joint geometry can significantly affect the effective stiffness of three-dimensional truss architectures, while Casata et al. [24] further confirmed the practical importance of joint-shape design in additively manufactured thin lattice structures. These studies support the view that joint geometry should be regarded as a secondary but potentially decisive design variable within a fixed topology and relative density.

2.3.3. Literature on Joint Geometry Effects

A growing body of literature suggests that joint geometry can significantly influence the behavior of lattice metamaterials, particularly when local bending, stress concentration, and nonlinear deformation are important [12, 18, 19, 24]. Studies introducing fillets, smoother transitions, or locally thickened connection regions have reported modified stiffness, delayed yielding, or altered collapse behavior compared to sharp or idealized geometries [18, 19, 24]. Latture et al. [18] showed, for an octet truss under compression, that nodal fillets modify the compressive response and that these effects interact with external boundary conditions. This is important because it demonstrates that joint-shape modifications are not merely local geometric refinements: they can influence the response of the full specimen.

Representative examples of joint-shape modification are shown in Figure 2.4, ranging from sharp and filleted joints to expanded and locally modified joint geometries.



a) Sharp Joint b) Filleted Joint c) Expanded/Rounded Joint d) Modified cross-section at the joint square, star

Figure 2.4: Representative joint-geometry modifications used in lattice design studies. (a) Sharp joint, author's schematic. (b) Filleted joint adapted from Casata et al. [24]. (c) Expanded/rounded joint, author's schematic. (d) Modified cross-section at the joint inspired partly by Portela et al. [12], showing examples such as square and star-like local joint sections. Together, these examples illustrate that joint geometry can be modified in several qualitatively different ways even when the surrounding lattice topology remains unchanged.

This constant-relative-density condition is also important. If the total amount of material increases together with joint smoothing or thickening, then an apparent performance improvement may simply reflect a heavier structure rather than a more efficient geometry. For this reason, studies that redistribute material between joints and struts while keeping $\bar{\rho}$ fixed are especially valuable, because they isolate the geometric effect more cleanly [12, 19, 24]. Portela et al. [12] showed that local geometric modifications in lattice connections can meaningfully affect the effective stiffness of three-dimensional truss architectures, while Casata et al. [24] demonstrated that nodal geometry optimization remains an active topic in additively manufactured metallic lattices. Doutre et al. [19] further showed that the effect of fillets in octet-truss lattices can change when mass is kept fixed, emphasizing the importance of distinguishing geometric improvement from simple material addition. Taken together, this literature indicates that connection geometry is not merely a secondary CAD detail, but a legitimate design variable whose effects can become comparable to other geometric parameters in some regimes.

Representative literature examples are summarized in Table 2.1, which highlights that the reported consequences of joint modification range from stiffness changes to altered specimen-level compressive response and substantial gains in modulus or yield strength under fixed-relative-density constraints.

Table 2.1: Representative literature examples showing how joint-geometry modification influences lattice response.

Reference	Topology	Joint modification	$\bar{\rho}$ controlled?	Response metric	Main reported effect
Portela et al. [12]	Octahedron, octet, tetrakaidecahedron, pyramidal	Circle / square / star node-contributing regions with constant strut area	Yes (approximately, within ±0.7%)	Effective stiffness	Joint geometry alters stiffness scaling; effect depends on rigid vs. non-rigid architecture
Latture et al. [18]	Octet truss	Nodal fillets	Not the primary focus	Compressive response	Fillets modify compressive behavior and interact with external boundary conditions
Casata et al. [24]	BCC; preliminary SC validation	Node rounding / fillet radius with reduced strut diameter	Yes	Elastic modulus; yield strength	Elastic modulus increased by 50.4–103.5%; yield strength increased by 26.4–45.7%; lower impact at low relative density
Doutre et al. [19]	Octet truss	Nodal fillets under identical-mass constraints	Yes	Effective modulus; yield strength; maximum stress	Fillets did not necessarily improve performance when the mass was kept fixed, because the struts had to be thinned to compensate for added fillet material

The existing body of work also suggests a methodological implication: if joint geometry is to be studied seriously as a design parameter, then the modeling framework must resolve the joint region with sufficient fidelity [12, 26, 27]. This immediately raises the question of how such fidelity can be retained without sacrificing the computational tractability needed for large-scale parametric studies.

2.3.4. Open Questions for Low-density Lattices

The motivation for this thesis arises from the hypothesis that joint geometry may remain an important design variable even in low-relative-density, stretching-dominated truss lattices, provided that its influence is studied under suitable geometric constraints. In particular, the present work investigates whether modifying the local cross-sectional stiffness near the joints, while preserving the global relative density and controlling the slenderness of the struts, can significantly alter the onset and development of nonlinear deformation.

Although prior work has shown that connection geometry can influence lattice response, important gaps remain. First, much of the existing literature focuses on density ranges, topologies, or loading conditions in which joint effects are easier to observe experimentally or numerically, while very low-relative-density regimes remain comparatively less explored. Second, systematic studies that isolate the influence of joint geometry at fixed relative density remain limited. Third, the specific combination of very low relative density, stretching-dominated truss architecture, fixed-density joint-geometry variation, nonlinear compression, and energy absorption remains only sparsely addressed in the available literature [12, 18, 19, 21, 24].

These open questions are difficult to address because both experiment and simulation become challenging in precisely the regime of interest. Glaesener et al. [17] showed that the response of periodic trusses is sensitive to geometric imperfections, while Portela et al. [26] argued for efficient modeling strategies precisely because detailed resolution of complex truss architectures becomes computationally demanding. This combination of physical relevance and computational difficulty is a major part of the motivation for the modeling strategy developed in this thesis.

2.4. Modeling Approaches for Lattice Metamaterials

The literature reflects a persistent modeling trade-off. On one end, full solid-element models can represent local geometry, stress gradients, and joint-driven nonlinear mechanisms with high fidelity. On the other, beam and truss models enable much larger simulations and broader parameter studies, but at the cost of idealizing the very regions where joint geometry is expected to matter most. Between these two extremes, a smaller body of work has explored mixed or reduced-order approaches that attempt to preserve local fidelity in the connection regions while keeping the remainder of the structure computationally affordable [12, 21, 24, 26, 27].

This section reviews beam- and truss-based idealizations in subsection 2.4.1, then full solid finite element approaches in subsection 2.4.2. Section 2.4.3 then discusses hybrid beam–3D continuum or reduced-order approaches that explicitly address the joint-resolution problem. Section 2.4.4 considers specimen size, localization, and boundary sensitivity in relation to model fidelity, before subsection 2.4.5 synthesizes the resulting trade-off between local accuracy and global scalability.

The three principal modeling families relevant to the present work are summarized schematically in Figure 2.5: beam/truss idealization, fully resolved solid finite element modeling, and hybrid beam–3D continuum discretization.

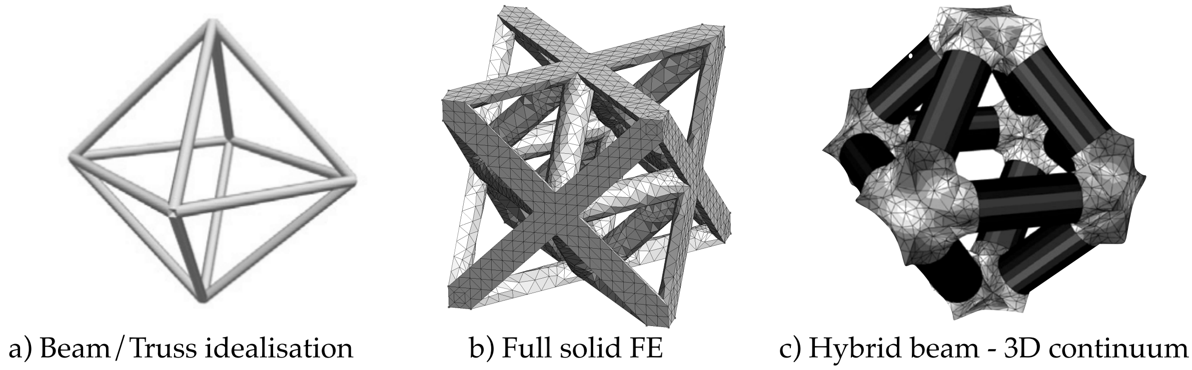


Figure 2.5: Schematic comparison of the principal modeling families discussed in this chapter. (a) Beam/truss idealization, author's own schematic. (b) Fully resolved solid finite element discretization, author's own schematic. (c) Hybrid beam–3D continuum discretization adapted from Portela et al. [12]. The figure highlights the central modeling trade-off of the present literature review: local geometric fidelity increases from left to right, while computational economy typically decreases.

Figure 2.5 provides a useful map for the following subsections, in which these three modeling families are compared in terms of accuracy, computational burden, and suitability for studying joint geometry.

2.4.1. Beam and Truss Idealizations

Beam and truss models are widely used in the simulation of lattice metamaterials because they reduce the structure to one-dimensional members connected through idealized joints [8, 10–12, 25]. This makes them computationally efficient and therefore well suited to large samples, parametric studies, topology screening, and studies where the dominant behavior is governed mainly by member-level mechanics [12, 25, 26].

Even when enriched with geometric nonlinearity, plasticity, or more advanced constitutive behavior, such models still rely on assumptions that become questionable when local connection geometry is expected to influence the response. In most beam-based descriptions, the joint behavior is either idealized as rigid or pinned, or represented through a prescribed effective stiffness rather than arising directly from geometry. Local stress concentrations, detailed transition shapes, and realistic joint-induced redistribution of stress are therefore not naturally resolved [12, 18, 21, 25].

Portela et al. [12] explicitly argued that simple beam-like and homogenized descriptions are effective mainly for slender architectures and do not capture the physics of non-slender truss architectures when junction effects become important. In parallel, Dong and Zhao [25] proposed a joint stiffening element to account for the influence of additively manufactured lattice joints on structural stiffness, thereby improving a beam-type representation without abandoning its one-dimensional character.

A further limitation of beam-only approaches in the present context is that they may fail specifically in the large-deformation regime of interest. Sun et al. [21] compared beam-element and solid-element simulations of additively manufactured lattice compression and found that the beam models did not converge under beam refinement, underpredicted post-yield stress levels, and delayed the onset of densification. They attributed these deficiencies in part to the inability of beam formulations to capture the true interconnection volumes at strut junctions and to represent contact between interconnected members accurately.

Taken together, these studies show that beam and truss models should not be regarded as universally inaccurate, but as limited in a specific and important way: their accuracy deteriorates when the response depends strongly on joint-region geometry, stress concentration, contact, or non-slender strut–joint transitions. In the present thesis context, this means that beam-only models remain valuable as a reference class and as a route to scalability, but they are not by themselves sufficient for answering the central research question on joint geometry effects.

2.4.2. Full Solid Finite Element Modeling

Full solid finite element models represent the actual three-dimensional geometry of both struts and joints. This allows them to capture local stress gradients, realistic transition shapes, buckling modes influenced by geometry, and the development of yielding or localization in the joint region [12, 21, 24]. For studies focused on local mechanics, they therefore offer a much higher level of physical fidelity than beam-based idealizations.

However, this fidelity comes at significant computational cost. Low-relative-density lattices contain slender struts that require sufficiently fine meshes to capture bending, instability, and stress gradients. The joints require additional local refinement to resolve the geometric transitions where stress concentrations occur. Moreover, a representative specimen may require many repeated unit cells, especially when nonlinear localization or boundary effects are important. As a result, full solid models often become too expensive for broad parametric sweeps across relative density, topology, and joint design [12, 21, 24, 26].

Portela et al. [12] used fully resolved three-dimensional finite element unit-cell models as reference solutions when quantifying the effect of joint geometry on stiffness scaling. Casata et al. [24] employed detailed finite element analysis on $3 \times 3 \times 4$ lattice specimens using quadratic tetrahedral elements to investigate the elasto-plastic response of node-rounded lattices. Sun et al. [21] meshed full $5 \times 5 \times 5$ lattice assemblies with second-order tetrahedral solid elements (C3D10M), after a mesh convergence study that led to a 0.5 mm mesh size, and noted explicitly that this resulted in a large number of elements and high computation cost.

The computational burden of fully resolved continuum modeling is driven by the combined need to discretize every strut and joint in three dimensions, to refine the mesh through slender members sufficiently to capture bending and instability, and to increase specimen size when finite-boundary or localization-sensitive behavior is of interest [12, 21, 24, 26]. As a result, the number of global unknowns, the memory usage, and the solution time all increase rapidly in precisely the parameter regime most relevant to this thesis. Fully solid models are therefore best understood as a high-fidelity benchmark class: indispensable when local mechanics must be resolved in detail, but often poorly suited to wide nonlinear parametric studies of large low-density lattice tessellations.

Illustrative examples from the current literature are compiled in Table 2.2. Even this small set is enough to show that higher-fidelity continuum modeling rapidly becomes demanding when either full unit-cell resolution or finite multi-cell specimens are considered.

A simple order-of-magnitude extrapolation further clarifies why this matters. Portela et al. [12] reported that the fully resolved validation unit cells required approximately 10^6 to 5×10^6 DOFs depending on node geometry, whereas the corresponding reduced-order unit cells reproduced the fully resolved stiffness to within 3% while reducing the problem size to only 0.02% to 0.3% of the original. If one were to extend a fully resolved unit-cell discretization of that type to a finite $5 \times 5 \times 5$ tessellation by naive replication, the total problem size would grow by roughly a factor of 125, leading to an order-of-magnitude estimate of 1.25×10^8 to 6.25×10^8 DOFs. Although such a calculation ignores differences in meshing strategy and solver implementation, it captures the essential point: explicit three-dimensional resolution of every strut and joint quickly becomes prohibitive at the specimen sizes relevant for finite-boundary and localization-sensitive studies.

2.4.3. Hybrid and Reduced-order Joint-resolving Approaches

Between beam-only and fully solid discretizations lies a smaller but particularly relevant class of methods: approaches that attempt to preserve high fidelity in the joint regions while avoiding a uniformly three-dimensional discretization of the entire lattice [12, 26, 27]. These methods are of direct interest to the present thesis because they address the same central tension between local accuracy and global scalability.

A notable contribution in this direction is the work of Portela et al. [12], who introduced reduced-order models for non-slender truss lattices in which the mechanically complex joint regions are represented by three-dimensional solid substructures, while the connecting gauge sections are modeled with Timoshenko beam elements. The motivation is explicit: a pure beam model fails to capture the response of non-slender lattices, while a full continuum model is too expensive to apply to finite tessellations of practical interest. Portela et al. [26] later extended this line of work and positioned it within a broader strategy for efficient simulation of complex periodic truss architectures.

The essential idea is to treat each joint as a local three-dimensional substructure and then reduce its internal degrees of freedom before assembling the full lattice model. If the degrees of freedom of a node substructure are partitioned into retained interface DOFs u_r and internal DOFs u_i, then the linear equilibrium system of the substructure can be written in block form as

$$\begin{bmatrix} K_{rr} & K_{ri} \\ K_{ir} & K_{ii} \end{bmatrix} \begin{bmatrix} u_r \\ u_i \end{bmatrix} = \begin{bmatrix} f_r \\ 0 \end{bmatrix}. \tag{2.3}$$

Eliminating the internal DOFs by static condensation gives the reduced relation

$$\left(K_{rr} - K_{ri} K_{ii}^{-1} K_{ir}\right) u_r = f_r, \tag{2.4}$$

so that only the interface behavior of the solid node needs to be retained at the global level. In the Portela formulation, six DOFs are retained at each beam–node connection face, namely three translations and three rotations, so that a node connected to z struts becomes an effective $6z$-DOF element instead of a tetrahedral mesh with thousands of internal DOFs.

This construction preserves the local three-dimensional stiffness of the joint while allowing the struts to remain one-dimensional beam members. The retained interface DOFs are then coupled to the beam DOFs in an averaged sense at each connecting face. For the octahedron and tetrakaidecahedron architectures studied by Portela et al. [12], this strategy reduced the problem size to 0.02%–0.3% of the fully resolved problem while keeping the maximum unit-cell stiffness error within 3%. This makes the reduced-order formulation attractive not only as a computational device, but as evidence that joint-resolving discretizations can remain mechanically credible at much lower cost than full solid models.

Conceptually, this type of discretization is attractive because the mechanically complex part of the architecture, namely the joint region, is given higher fidelity, while the more slender strut-like portions

Table 2.2: Illustrative literature examples showing the discretization burden of higher-fidelity continuum-based lattice models.

Reference	Topology / geometry	Specimen size	Model class	Reported mesh or DOF (degree of freedom) information	Nonlinearity	Main purpose
Portela et al. [12]	Variable-node rigid and non-rigid unit cells	$1 \times 1 \times 1$ unit cell with PBC (periodic boundary conditions)	Fully resolved 3D reference FE	10^5 to 10^6 tetrahedral elements and approximately 10^6 to 5×10^6 DOFs, depending on node geometry	Linear elastic	Reference stiffness scaling study free of finite-boundary effects
Portela et al. [12]	Node substructures for reduced-order hybrid model	Node substructures within $5 \times 5 \times 5$ lattices	3D node FE before condensation	Octahedron nodes: 15 000–27 000 DOFs condensed to 48 DOFs; tetrakaidecahedron nodes: 8 000–12 000 DOFs condensed to 24 DOFs	Linear elastic	Quantify benefit of node condensation in hybrid reduced-order model
Sun et al. [21]	Octet, RD, HS polymer lattices	$5 \times 5 \times 5$ unit cells	Full specimen 3D FE (C3D10M)	Converged 0.5 mm mesh; authors note large element count and high computation cost; explicit solver with mass scaling used	Material nonlinearity, damage, large deformation, contact	Compression, energy absorption, anisotropy, deformation localization
Casata et al. [24]	BCC and SC lattices with rounded nodes	$3 \times 3 \times 4$ unit cells	Full specimen 3D FE (Tet10)	Quadratic tetrahedral mesh; element size 0.2 mm for BCC-H and 0.08 mm for BCC-L; estimated total problem size of order 10^7 to 3×10^7 DOFs for full-specimen analysis	Elasto-plastic	Quasi-static compression and node-rounding optimization under fixed relative density

are treated in a reduced way. This idea is especially important for low-relative-density lattices. In a uniformly solid discretization, decreasing relative density leads to thinner and more slender struts, which tends to increase mesh-resolution demands and therefore computational cost. In contrast, a hybrid beam–3D continuum discretization can avoid meshing the entire strut volume with solid elements, so the overall number of degrees of freedom does not grow in the same way as relative density decreases. Even before considering additional model-order reduction, the discretization choice itself therefore offers a potentially favorable scaling behavior.

The quantitative benefit of this strategy is already visible in the linear studies of Portela et al. [12]. In their reduced-order formulation, node substructures with many thousands of internal DOFs were statically condensed to a small set of interface DOFs, giving each node an effective $6z$-DOF description, where z is the nodal connectivity. For the geometries considered, the condensed models remained within a maximum stiffness error of 3% relative to the fully meshed unit-cell problem while reducing the problem size to 0.02% to 0.3% of the original one. This strongly suggests that the underlying discretization philosophy is not only computationally attractive but also mechanically credible, and therefore worth extending toward the nonlinear regime addressed in this thesis.

Related joint-resolving model-reduction strategies have also been proposed. For example, De Weer et al. [27] introduced a parametrized superelement approach for lattice joint modeling and simulation, which similarly reflects the need to retain joint-level fidelity without resolving the full lattice as a monolithic continuum model.

From the perspective of the present thesis, this type of approach is appealing for two reasons. First, it is more physically compatible with the hypothesis that joint geometry matters, because it resolves the region where that effect originates. Second, it remains closer in spirit to scalable lattice analysis than a fully solid model. This makes it a natural methodological starting point for a nonlinear-compatible study of joint geometry in very low-relative-density truss lattices.

This also distinguishes hybrid beam–3D continuum approaches from reduced beam formulations with equivalent joint corrections, such as the joint stiffening element proposed by Dong and Zhao [25], where joint effects are incorporated indirectly rather than through an explicitly discretized continuum joint region.

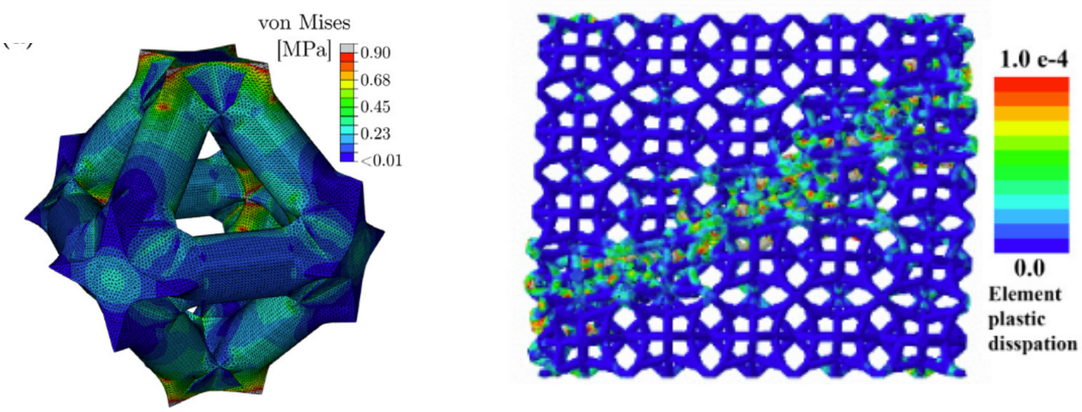
2.4.4. Representative Specimen Size, Localization, and Boundary Effects

The question of model fidelity is not limited to element type or local geometry representation. It also depends on specimen size. In lattice metamaterials, the number of unit cells included in a simulation can directly influence the observed response, especially once the focus shifts from effective elastic properties to progressive collapse, localization, and boundary-sensitive nonlinear behavior [12, 21, 24, 26].

At one end of the spectrum, single unit-cell models with periodic boundary conditions are highly useful for extracting effective properties and for studying idealized architecture-dependent scaling trends [8, 10–12]. However, they remove free-boundary effects and cannot represent finite-size localization patterns or specimen-level collapse modes. For this reason, they are insufficient when the research question concerns the development of local failure within a finite lattice specimen [12, 21, 24].

The difference between periodic unit-cell analysis and finite-specimen analysis is illustrated in Figure 2.6.

Several studies therefore use finite tessellations rather than single-cell periodic models. Portela et al. [12] used finite lattice samples to capture boundary-sensitive behavior, while Portela et al. [26] continued this direction in the context of efficient modeling of complex periodic truss architectures. Sun et al. [21] also studied finite lattice specimens and showed that the deformation response can be linked to the number of cell layers in the sample; in particular, the octet response exhibited five stress peaks corresponding to the five cell layers of the $5 \times 5 \times 5$ specimen. Casata et al. [24] likewise used finite specimens for comparative analysis of joint modifications in thin lattice structures. These examples are



a) 1 Unit Cell with periodic BC's b) Finite 5x5x5 bending-dominated specimen

Figure 2.6: Comparison between periodic unit-cell analysis and finite-specimen analysis. (a) Periodic unit-cell stress field under linear compression, adapted from Fig. C.7(a) of Portela et al. [12]. (b) Element plastic dissipation field in a finite $5\times5\times5$ bending-dominated specimen under localized crushing, adapted from Fig. 20 of Sun et al. [21]. The figure highlights that periodic unit-cell models are appropriate for extracting idealized effective response, whereas finite specimens are required to capture localization and specimen-scale failure patterns.

summarized in Table 2.3.

Table 2.3: Representative specimen sizes and modeling roles reported in the literature.

Reference	Topology	Model class	Specimen size	Periodic / finite	Main phenomenon studied
Portela et al. [12]	Octahedron, tetrakaidecahedron	Hybrid reduced-order FE (3D node substructures + beam struts)	$5\times5\times5$	Finite	Effective stiffness of typical experimental samples; boundary effects retained
Portela et al. [26]	Non-slender truss architectures	Reduced-order nodes + QC (quasicontinuum) framework (outlined)	$5\times5\times5$ example samples; larger QC lattices discussed	Finite and large-scale periodic QC setting	Efficient modeling of non-slender lattices and path toward larger nonlinear tessellations
Sun et al. [21]	Octet, RD, HS	FE cell assemblies + experiments	$5\times5\times5$	Finite	Compression, progressive layer-wise crushing, deformation-mode dependence on loading direction
Casata et al. [24]	BCC; SC validation	Full specimen 3D FE + experiments	$3\times3\times4$	Finite	Elasto-plastic compression and comparative effect of node rounding

The correct conclusion is therefore not that there exists one universal minimum specimen size, but rather that model size must be chosen in relation to the phenomenon being studied. Effective-property extraction, first instability, progressive crushing, and localization propagation may all require different specimen extents. For the present thesis, this point is important because joint-driven nonlinear mechanisms are inherently local, while the resulting collapse behavior is expressed at the specimen scale. A meaningful modeling framework must therefore be able to represent both [12, 21, 24].

2.4.5. The Modeling Trade-off: Scalability versus Local Fidelity

The practical modeling gap is therefore not simply one of "accuracy versus speed", but more specifically one of capturing joint-driven nonlinear mechanisms at a cost that still allows systematic design-space exploration. The regions of the structure that matter most for the present research question are exactly those that beam and truss models simplify most aggressively, while the approaches that resolve those regions most faithfully are often too expensive to use on large, low-density lattice samples [12, 21, 24, 26, 27].

This issue becomes particularly severe in low-relative-density truss metamaterials. As the relative density decreases, the struts become more slender, the response becomes more sensitive to instability and localization, and the number of unit cells required to observe representative structural behavior may increase [8, 10, 17, 21]. The regime in which joint effects are most interesting is therefore also the regime in which model size and computational cost increase most rapidly [12, 21, 24, 26].

This tension is reflected directly in the literature. Sun et al. [21] and Casata et al. [24] illustrate the value of high-fidelity continuum modeling for nonlinear and joint-sensitive problems, but also the natural computational burden of such analyses. Portela et al. [26] argued for efficient modeling approaches precisely because complex periodic truss architectures become computationally demanding to analyze in detail, while Portela et al. [12] showed that carefully condensed joint-aware models can preserve high accuracy in the linear regime at vastly reduced problem size. Taken together, these works highlight the central methodological tension addressed in this thesis.

Hybrid beam–3D continuum approaches are therefore attractive not merely because they sit midway between two classical modeling extremes, but because they align more closely with the spatial structure of the physical problem itself. The joint regions, where local geometric detail matters most, can be treated with higher fidelity, while the struts, which mainly transmit load over longer distances, can be represented more economically.

These observations motivate the search for a modeling framework that retains sufficient local fidelity in the joint regions while remaining scalable enough for broader nonlinear parametric studies. This is the methodological gap that the present thesis seeks to address.

2.5. Research Gap and Thesis Positioning

The literature reviewed above suggests that low-relative-density truss lattices are an especially attractive but difficult class of architected metamaterials. Their performance is strongly shaped by topology and relative density, yet their nonlinear response may also depend in important ways on local joint geometry. At the same time, the computational approaches best suited to large-scale parametric studies tend to idealize the joints, while the approaches best suited to resolving joint geometry remain difficult to scale to the specimen sizes and parameter ranges of interest [12, 21, 24, 26, 27].

This final section summarizes the resulting gap in subsection 2.5.1 and then positions the contribution of the present thesis in subsection 2.5.2.

2.5.1. Gap in the Literature

A clear gap emerges at the intersection of three observations. First, the low-relative-density regime is one of the most attractive for lightweight structural applications and therefore one of the most important to understand. Deshpande et al. [10] established the particular structural efficiency of stretching-dominated architectures, while Zheng et al. [13] and Meza et al. [14] showed that architected materials can achieve exceptional specific mechanical performance at very low density. Yin et al. [22] further highlighted the continuing relevance of lattice structures in the energy absorption literature.

Second, this same regime is particularly sensitive to local geometric effects, instability, and nonlinear deformation, which suggests that joint geometry may act as a practically relevant design variable. Portela et al. [12] demonstrated that joint geometry can influence the effective stiffness of truss lattices, while Latture et al. [18], Casata et al. [24], and Doutre et al. [19] showed that changes in joint or nodal geometry can influence compressive response, stiffness, strength, or the interpretation of performance under fixed-density constraints. In addition, Glaesener et al. [17] showed that geometric imperfections can significantly alter periodic truss response.

Third, the available modeling approaches do not easily allow systematic investigation of these effects. Beam-based models scale well but simplify the joints, whereas full solid models resolve the joints but scale poorly. Portela et al. [26] therefore proposed efficient modeling approaches for complex periodic truss architectures, highlighting the practical need for methods that balance fidelity and cost. The quantitative examples in Table 2.2 and Table 2.3 make this tension more explicit: the relevant finite-specimen problems are large enough to expose boundary-sensitive and localization-driven behavior, but this is precisely what makes full continuum modeling difficult to scale.

A more specific methodological gap follows from this. Existing literature indicates that joint geometry matters, that fixed-relative-density treatment is important when interpreting joint modifications, and that approaches resolving the joint region can be substantially more informative than pure beam idealizations in joint-sensitive architecture classes [12, 18, 19, 24]. However, scalable nonlinear-compatible frameworks for systematically studying how joint geometry affects the response of low-relative-density truss metamaterials under controlled geometric constraints remain limited.

In other words, the gap is not merely a lack of data on joint design. It is a lack of modeling capability suited to generating such data efficiently and credibly, while keeping the remaining model error explicit.

2.5.2. Scope and Contribution of this Thesis

Motivated by this gap, this thesis focuses on the computational investigation of low-relative-density truss metamaterials with explicit attention to joint geometry. The central aim is not merely to reproduce known topology-dependent trends, but to enable systematic study of how modified joint regions influence nonlinear response in regimes where instability, localization, and post-yield behavior are important.

To address this need, this thesis develops and evaluates a nonlinear-compatible hybrid beam–3D continuum modeling approach for truss lattice metamaterials. In this strategy, the joint regions are discretized with three-dimensional continuum elements in order to retain local geometric fidelity, while the slender struts are represented with beam elements to preserve computational scalability. The method is positioned between the two dominant modeling classes identified in the literature: fully solid finite element models, which provide high local fidelity but are expensive to scale, and beam-based lattice models, which are computationally efficient but idealize the joint regions.

The main contribution of the thesis is methodological. It investigates whether a joint-resolving hybrid discretization can provide a practical intermediate modeling strategy for studying joint-geometry effects in low-relative-density truss lattices without resorting to a uniformly solid discretization of the entire specimen. In doing so, the thesis establishes the formulation, solver strategy, validation procedure, and accuracy assessment needed to judge the usefulness and limitations of this approach.

3

Governing Equations and Modeling Assumptions

Lattice metamaterials across different architectures, length scales, and geometric parameter variations share a common structural feature: they are composed of slender members connected through joints. This common topology has motivated a wide range of modeling approaches that reduce computational cost while retaining the mechanical accuracy needed to describe member deformation, joint effects, and their interaction. These approaches range from pure beam models with different beam formulations to hybrid or reduced-order methods that incorporate a more explicit representation of the joint regions [12, 26].

This chapter presents the computational framework developed and used in this thesis. Section 3.1 defines the material assumptions, modeling scope, and non-dimensional unit system used throughout the thesis. Section 3.2 introduces the lattice architectures relevant to this work, while section 3.3 defines the geometrical parameters used to describe them. Section 3.4 then introduces the element types used in the thesis, namely beam elements and solid continuum elements, together with the assembly strategy and solver choices used for standalone beam and solid unit-cell models. The hybrid beam–3D continuum modeling approach is developed in section 3.5, including the interface coupling strategy and main implementation design.

3.1. Modeling Scope, Non-Dimensionalization and Material Assumptions

The simulations in this thesis are formulated in a non-dimensional unit system. The unit-cell length is taken as the reference length, so geometric dimensions are expressed relative to $L_{\mathrm{UC}} = 1$, and strut thickness is controlled through the ratio r/l. This choice removes dependence on a particular physical specimen size and allows direct comparison between lattice architectures, joint geometries, and modeling approaches. Reported unitless effective stresses, moduli, and energy-like quantities should therefore be interpreted as normalized comparative quantities rather than dimensional material properties.

A single isotropic elastic material is used throughout the report, with Young's modulus $E = 10000$ and Poisson's ratio $\nu = 0.3$ in the non-dimensional unit system. Since the same material parameters are used consistently across the solid, hybrid, and beam-only models, the absolute choice of stiffness scale does not affect the relative comparison between modeling approaches. The quantities reported in later sections are therefore used to compare model and metamaterial geometry behavior.

The main simulations use an elastic material model with geometrically nonlinear kinematics. The nonlinear response studied in the validation and accuracy chapters therefore arises from large deformation, large rotation, buckling, post-buckling deformation, and load-path redistribution, rather than from plasticity, damage, fracture, or material degradation. The models are thus materially elastic but geometrically nonlinear.

This modeling scope is intentional. By excluding irreversible constitutive effects from the main validation and accuracy studies, the influence of joint geometry and modeling idealization can be isolated more clearly. The resulting effective stress–strain nonlinearity should therefore be interpreted as the nonlinear structural response of elastic lattices, not as a prediction of permanent material failure.

3.2. Lattice Architectures Relevant to This Work

This section identifies the lattice architectures used in the remainder of this thesis and explains why they were selected. The general distinction between bending-dominated and stretching-dominated architectures, together with representative unit-cell schematics, has already been introduced in chapter 2; in particular, the relevant topologies are shown in Figure 2.1. The purpose of the present section is to specify which architectures are used in the modeling framework developed in this work and what role each of them plays in the later chapters.

The selected architectures span both bending-dominated and stretching-dominated deformation regimes. This is important because the sensitivity of lattice response to joint modeling is expected to depend not only on relative density and nonlinearity, but also on the rigidity class of the underlying topology. By considering multiple architectures, the later studies can therefore distinguish between effects that are topology-specific and effects that are more general to the hybrid beam–3D continuum modeling approach.

The architectures considered in this thesis are summarized in Table 3.1. In all cases, the lattice is constructed by periodic repetition of a unit cell composed of straight struts meeting at joints. The specific arrangement and connectivity of those struts determine whether the architecture is primarily bending-dominated or stretching-dominated.

Table 3.1: Lattice architectures considered in this thesis and their role in the subsequent chapters.

Architecture	Deformation class	Unit-cell construction	Role in this thesis
BCC	Bending-dominated	The body-centered cubic unit cell is formed by struts connecting the cube corners to a central joint. The load path is therefore not purely axial, and the architecture admits significant bending contributions under macroscopic loading.	Used as a representative bending-dominated architecture for studying how joint modeling affects stiffness, instability, and nonlinear response in a topology that is sensitive to local geometric detail.
Pyramidal	Bending-dominated	The pyramidal unit cell is formed by inclined struts connecting a base plane to an apex joint, creating a topology with lower connectivity and strong bending contributions under compression or shear.	Used as an additional bending-dominated comparison case, allowing the influence of joint geometry to be examined in a topology structurally distinct from BCC while remaining non-rigid in the sense discussed in chapter 2.
Octet	Stretching-dominated	The octet-truss unit cell is formed by a highly connected network of diagonal and edge-aligned struts arranged so that loads are transferred primarily through axial stretching and compression rather than member bending.	Used as a representative stretching-dominated architecture, providing a contrast with BCC and pyramidal lattices and allowing assessment of whether joint-sensitive effects remain important in a topology with efficient axial load paths.

The selection of BCC, pyramidal, and octet lattices therefore provides a compact but mechanically meaningful set of case studies. Bending-dominated and stretching-dominated behavior are both represented, while the underlying unit-cell constructions remain simple enough to be parameterized consistently within the modeling framework. This makes the set well suited for later validation, comparison, and nonlinear studies of joint geometry effects.

3.3. Definitions of Geometrical Parameters for 3D Lattices

As explained previously, the geometry is described using non-dimensional parameters. For each architecture, the unit-cell side length is denoted by L, while l denotes the characteristic strut length. In architectures where all struts have equal length, l is simply that common member length. Architectures with multiple strut lengths l are not considered in this work.

The global lattice size is defined by the number of repeated unit cells in the three coordinate directions,

$$N_x, \qquad N_y, \qquad N_z.$$

The main local slenderness parameter is the strut radius-to-length ratio,

$$\frac{r}{l},$$

where r is the radius of the circular strut cross-section away from the joint region. This ratio is used throughout the thesis as the primary geometric proxy for relative density and strut slenderness.

These parameters are illustrated in Figure 3.1, where a $5 \times 5 \times 5$ BCC lattice is shown together with the N_x, N_y, and N_z directions. Figure 3.3 shows pyramidal unit cells over a range of r/l values.

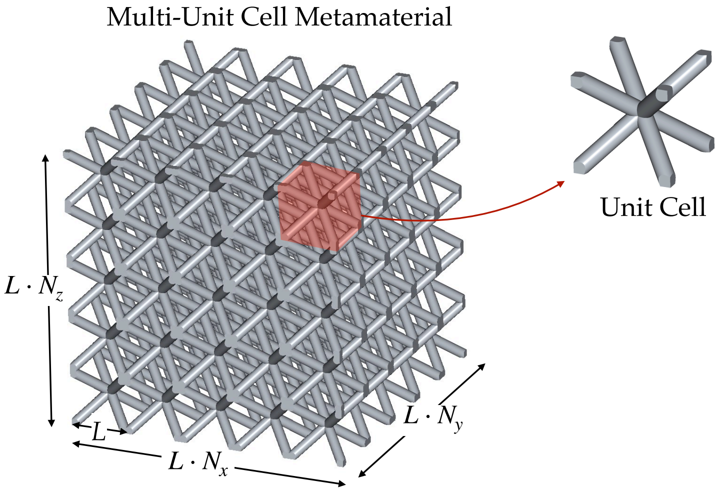
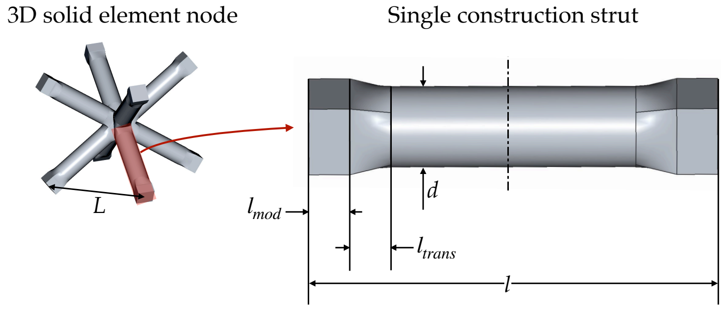


Figure 3.1: Render of a $5 \times 5 \times 5$ BCC lattice metamaterial, showing the N_x, N_y, and N_z unit-cell count parameters.

In addition to these global lattice parameters, the local parameters are required to describe the modified joint geometries used in this thesis. Figure 3.2 shows a detail of a single strut within a BCC unit cell, including the joint geometry generation parameters l, d, l_{mod}, and l_{trans}, where d is the strut diameter. These parameters define geometries that vary the moment of inertia along the strut axis near the joint while keeping the modified cross-sectional area constant. This allows the effect of local cross-sectional shape to be studied as independently as possible from changes in relative density.

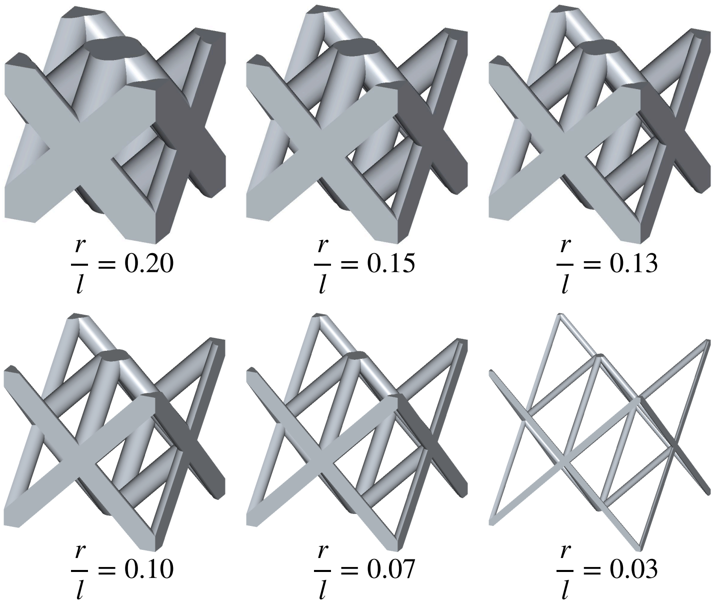
To remain consistent with the approach of Portela [12], two geometries are retained: a circular baseline geometry and a square modified geometry. In both cases, a region of the strut near the joint is assigned a modified cross-section before transitioning back to the circular strut geometry away from the joint. The local joint geometry is described by the following parameters:

- modified cross-section length from the joint center along the strut axis: l_{mod},


3D solid element node

Single construction strut

Figure 3.2: Detail of a single strut within a BCC unit cell, showing the strut length l, diameter d, modified-section length l_{mod}, and transition length l_{trans} used to design the modified square joint geometry.


$\frac{r}{l} = 0.20$ $\frac{r}{l} = 0.15$ $\frac{r}{l} = 0.13$

$\frac{r}{l} = 0.10$ $\frac{r}{l} = 0.07$ $\frac{r}{l} = 0.03$

Figure 3.3: Pyramidal unit cells with circular joint geometry over a range of r/l values from 0.20 to 0.03.

- length of the transition region from the modified cross-section to the circular strut cross-section: l_{trans},
- modified cross-section shape, defined by the parameters specific to each geometry.

To keep the joint-geometry definition consistent across different r/l ratios, these lengths are non-dimensionalized by the strut radius $r = d/2$, giving

$$\frac{l_{\text{mod}}}{r}, \qquad \frac{l_{\text{trans}}}{r}.$$

The joint geometries are shown in Figure 3.4. The upper renders show the local modified cross-section and transition region on an individual strut, while the lower renders show the corresponding pyramidal joint geometry. The circular geometry is used as the baseline case, and the square geometry is used as the modified case. The square modified section is defined so that its area matches that of the circular baseline section, and both cross-sections share the same centroid on the strut axis.

Following the choice made by Portela[12], the non-dimensional modified cross-section length is set to $l_{\text{mod}}/r = 1$, and the non-dimensional transition length is set to $l_{\text{trans}}/r = 1.5$ for both geometries. This choice ensures that the non-circular modified section does not occupy more than 75% of the strut length in the least slender hybrid-model case considered in this work, namely $r/l = 0.15$. The transition between the modified section and the circular strut was generated in FreeCAD using the `makeLoft` routine, which creates a face, shell, or solid by interpolating between two or more cross-sectional profiles. As a result, the circular and square geometries were designed to be equal in volume, but the final CAD geometries are only approximately so.

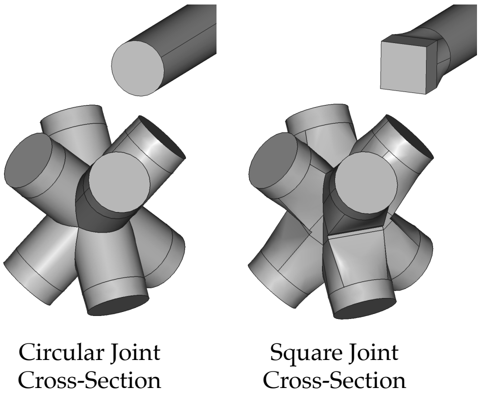


Figure 3.4: Joint geometries retained from Portela [12] in this thesis. Left: circular baseline geometry. Right: square modified geometry. The upper renders show the local modified strut cross-section and transition region next to the corresponding pyramidal joint geometry.

To quantify the remaining CAD-level mismatch, the solid volumes of pyramidal unit-cell models were compared for the circular and square joint geometries at two representative values of r/l. The results are given in Table 3.2.

Table 3.2: CAD volume comparison between circular and square joint geometries for pyramidal unit-cell models.

Model	r/l	V_{circ}	V_{sq}	$V_{\text{sq}}/V_{\text{circ}}$
Pyramidal unit cell	0.03	0.02641810	0.02643819	1.000760
Pyramidal unit cell	0.10	0.34395439	0.34469823	1.002163

As shown in Table 3.2, the square-to-circular volume ratio deviates from 1 by less than 0.22% for the cases examined here. These differences are considered insignificant in the present work, since

FEA (finite element analysis) meshing ultimately modifies the exact geometries further. No additional geometry-generation correction was introduced to enforce exact volume equality. The circular and square joint geometries are therefore treated as effectively equal-volume cases, representative of the same relative-density class of lattice metamaterial.

As discussed in chapter 2, several alternative joint geometries have been studied in the literature. Geometries containing internal voids are excluded from the present thesis because their behavior can depend on the manufacturing method given residual material remains trapped inside the voids. Excluding these cases keeps the study focused on joint geometries that can be interpreted more generally across manufacturing processes.

Filleted joint geometries are also excluded from the main parameter set. Fillets are important in practical design because they reduce local stress concentrations at sharp transitions. However, the purpose of this thesis is to isolate the effect of cross-sectional shape and moment-of-inertia variation near the joint. Including fillet radius as an additional design parameter would make it harder to separate this effect from the broader influence of stress-concentration reduction. Since no attention is paid to stress concentrations, or material models with stress concentration sensitivity in this thesis, the lack of sharp-corner refinement is not expected to affect the main conclusions about the influence of local joint shape on the global elastic response.

The hybrid modeling framework developed in section 3.5 is nevertheless more general than the two joint geometries studied explicitly. Once a lattice architecture is defined, any joint geometry can be introduced provided that a parametric CAD model and finite element mesh can be generated for the joint region. The circular and square geometries therefore serve both as physically motivated case studies and as demonstrations of the ability of the method to incorporate geometry-specific joint models.

3.4. Element Types in this Thesis

The modeling framework used in this thesis combines two finite element idealizations. Beam elements are used to represent slender lattice struts efficiently, while solid continuum elements are used where three-dimensional joint deformation must be resolved. Additionally, beam-only element and solid-only element models are used for baseline comparisons. This section introduces both element types and explains how pure beam and pure solid models are assembled before the hybrid coupling formulation is introduced in section 3.5.

3.4.1. Beam Elements

Beam elements are used throughout this thesis to represent the slender struts of lattice metamaterials at low computational cost. In all beam-based formulations, each beam node carries six degrees of freedom: three translations and three rotations, ordered as

$$\mathbf{q}_a = [u_x, u_y, u_z, \theta_x, \theta_y, \theta_z]_a^{\mathsf{T}}.$$

A two-node beam element therefore carries a 12-component nodal state vector.

For the pure beam lattice models used as the main reduced-order baseline in this thesis, and for the beam submodels used in the hybrid simulations, the formulation is the corotational beam implementation available in the Summit solver [28]. The solver supports both Euler–Bernoulli and Timoshenko beam theory. Both formulations are used in the pure beam comparisons, while the hybrid beam–3D continuum models use the Timoshenko formulation so that shear deformation can be represented in the beam struts.

In the corotational framework, rigid-body motion is separated from deformational motion by introducing a moving element frame based on the current nodal positions. The element kinematics are expressed in this corotated frame, where strains remain small even when the structure undergoes large global rotations. Internal element resultants are then evaluated in the corotated frame and rotated back to the global frame for assembly. In compact form,

$$\mathbf{f}_{\text{int}} = \mathbf{R}_e\,\hat{\mathbf{f}}_{\text{int}}(\hat{\mathbf{q}}), \qquad \hat{\mathbf{q}} = \mathbf{R}_e^{\mathsf{T}}\mathbf{q},$$

where $\mathbf{R}_e$ is the element corotational transformation. This is the main reason why the beam formulation remains effective for lattice deformation problems involving large rigid-body rotations of the struts.

The beam models developed in the present work use six-DOF beam end states and transmit axial forces, shear forces, bending moments, and torsional moments through the lattice network. Since the hybrid coupling formulation only requires consistent six-DOF end kinematics and six-component end resultants, the interface formulation developed later in this chapter is intentionally written independently of the internal beam weak form.

Some of the pure beam unit-cell models later introduced in this report were modeled in the OpenSeesPy Python finite element framework [29], which also provides beam-column formulations suitable for geometrically nonlinear structural analysis. The OpenSeesPy and Summit beam models used here represent the same intended beam idealization, so the pure beam models and the beam submodels in the hybrid simulations are compared at the level of the same reduced-order structural description.

3.4.2. Solid Continuum Elements

The joint regions are discretized with three-dimensional solid continuum elements using first-order continuous tetrahedral elements, denoted `Tet1CG` in Summit [28]. Unlike the beam formulation, the solid discretization carries only translational nodal degrees of freedom. Rotational interface quantities are introduced later only as rigid-face parameters used to prescribe or recover averaged motions on selected interface faces.

The tetrahedral meshes are generated from CAD joint geometries using Gmsh. In practice, the meshing process first discretizes the boundary surfaces of the joint and then fills the enclosed volume with unstructured tetrahedral elements subject to the chosen target element size and local geometric constraints. This makes the approach flexible for complex joint shapes, transition regions, and architecture-specific connection geometries, while preserving a straightforward export path to Summit-readable mesh files.

Two classes of boundary conditions are relevant in this thesis. Dirichlet boundary conditions prescribe kinematic quantities, such as displacements on solid nodes or rigid-face motions on classified solid faces. Neumann boundary conditions prescribe generalized surface actions, such as tractions or, after face integration, force and moment resultants. In the current hybrid implementation, coupled joint faces are driven kinematically, while unconstrained faces remain natural Neumann boundaries unless specific load-case prescriptions are applied.

The pure solid unit-cell simulations used later in this thesis are solved as monolithic geometrically nonlinear continuum problems with nodal displacements as the unknowns. At each load increment, the global residual and tangent stiffness are assembled over the full solid mesh and the equilibrium condition is enforced iteratively. For standard displacement-controlled unit-cell simulations, a damped Newton–Raphson strategy is used, with adaptive cutbacks of the imposed increment when a trial state fails to satisfy the acceptance criteria. When the response contains a limit point or a descending equilibrium branch, an arc-length scheme is used instead so that the solution path can be followed through the onset of instability and into the post-buckling regime.

3.4.3. Assembly of Pure Beam-Element Lattices

Pure beam models provide the baseline for all lattice simulations in this thesis. Each strut is represented by a number of equal-length beam elements, and the lattice is assembled as a network of beams connected at joints. At strut–strut connections, the model assumes full rigidity: beams meeting at a joint share the same translational and rotational degrees of freedom. This enforces displacement and rotation continuity and allows transfer of bending moments across the joint. Although alternate methods like torsional springs could be used to represent partial joint stiffness, the rigid-joint assumption is retained here for simplicity and most importantly, physical interpretability.

Global equilibrium of the beam lattice is enforced through a nonlinear solve. The assembled system may be written as

$$\mathbf{R}_{\text{beam}}(\mathbf{q}) = \mathbf{f}_{\text{int}}(\mathbf{q}) - \mathbf{f}_{\text{ext}} = \mathbf{0},$$

and is solved in Summit using the Newton–Raphson solver. At each iteration, element residuals and tangent contributions are assembled, boundary conditions are applied, and the nodal displacement and rotation field is updated until the residual norm falls below the prescribed tolerance. Finer load stepping is used when necessary to improve robustness under strongly nonlinear structural response. The full unit-cell beam models are solved in OpenSeesPy using a similar strategy, with the Newton-Raphson solver and same convergence criteria.

3.5. The Hybrid Beam–3D Continuum Modeling Approach

Solid element models can provide accurate predictions of local joint response and naturally accommodate complex constitutive behavior such as plasticity, fracture, viscoelasticity, and dynamics. However, their cost grows rapidly when large non-periodic lattice specimens must be resolved in full three dimensions. As the strut diameter-to-length ratio decreases, increasingly fine solid meshes are required to resolve the slender members, and the resulting global models become prohibitively expensive for large architectures.

Beam element models, presented in subsection 3.4.3, are computationally efficient and scale well to large structures, but they do not resolve the three-dimensional stress and deformation fields that develop inside lattice joints. Reduced-order approaches, such as those proposed by Portela, Greer *et al.* [12] and discussed in section 2.4, provide an attractive compromise by enriching beam models with precomputed nodal information. Their main limitation is that the precomputed reduction may cease to be valid once strong nonlinearities are introduced at the joint level. For geometric nonlinearity, plasticity, or fracture, the joint response must instead be recomputed on the current deformed configuration, together with the associated equilibrium iterations and constitutive updates.

The approach proposed in this thesis combines the accuracy of solid joint models with the efficiency of beam strut models while preserving compatibility with nonlinear modeling. Instead of replacing the joint by a precomputed linear reduced model, each joint remains a finite element subproblem that can be solved at the current state. The resulting hybrid model preserves a clear separation between submodels. This modularity makes surrogate replacement possible in principle, although surrogate construction and validation are outside the scope of the present thesis. This section presents the notation, coupling strategy, and formulation choices developed in this work. Section 3.5.1 introduces the solid–beam interface variables and kinematics, while Section 3.5.3 motivates the partitioned iterative strategy adopted later in the chapter. Table 3.3 repeats a summary of the principal symbols used in this section for reference.

3.5.1. Interface Coupling Strategy Between Solid and Beam Elements

The interface between a three-dimensional solid joint and a one-dimensional beam strut is treated as a low-dimensional coupling surface on which only a small set of generalized variables is exchanged. In this work, each coupling interface is denoted by Γ_i and consists of one classified solid face patch paired with one beam end. On the beam side, the end motion is described by a 6-component vector

$$\mathbf{q}_{\Gamma i}^{(b)} = \begin{bmatrix} \mathbf{t}_{\Gamma i}^{(b)} \\ \boldsymbol{\theta}_{\Gamma i}^{(b)} \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z & \theta_x & \theta_y & \theta_z \end{bmatrix}_{\Gamma i}^{\mathsf{T}},$$

Table 3.3: Principal notation used throughout the hybrid formulation section

Symbol	Meaning
Ω_s	Solid-joint domain.
Ω_b	Beam-strut domain.
Γ_i	Coupling interface i, pairing one solid face patch with one beam end.
$\mathbf{u}^{(s)}(\mathbf{x})$	Solid displacement field at position $\mathbf{x}$ on a joint.
$\mathbf{q}_a$	Six-component beam nodal degree-of-freedom vector at beam node a.
$\mathbf{q}_{\Gamma i}^{(s)}$	Six-component rigid-face motion used to describe interface Γ_i on the solid side.
$\mathbf{q}_{\Gamma i}^{(b)}$	Six-component beam-end motion at interface Γ_i.
$\mathbf{q}_{\Gamma i}$	Common interface motion once compatibility is imposed.
$\mathbf{f}_{\Gamma i}^{(s)}, \mathbf{f}_{\Gamma i}^{(b)}$	Six-component interface resultants returned by the solid and beam submodels.
$\mathbf{r}_{\Gamma i}$	Interface residual at Γ_i.
$\mathbf{q}$	Global stacked interface vector.

where $\mathbf{t}_{\Gamma i}^{(b)}$ contains the three beam-end translations and $\boldsymbol{\theta}_{\Gamma i}^{(b)}$ contains the three beam-end rotations. Figure 3.5 shows the role of this interface state within the hybrid lattice decomposition.

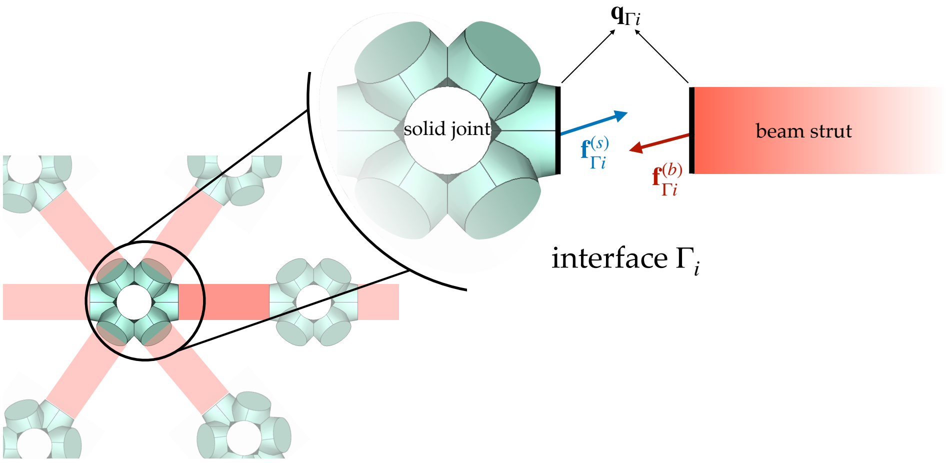


Figure 3.5: Hybrid decomposition around one interface Γ_i. The common interface motion $\mathbf{q}_{\Gamma i}$ is exchanged between the solid joint and the beam strut, while each submodel returns its own interface resultant, $\mathbf{f}_{\Gamma i}^{(s)}$ or $\mathbf{f}_{\Gamma i}^{(b)}$. Equilibrium is enforced by driving the residual $\mathbf{r}_{\Gamma i} = \mathbf{f}_{\Gamma i}^{(s)} + \mathbf{f}_{\Gamma i}^{(b)}$ to zero.

Also seen in figure 3.5, on the joint side, the same interface is represented by a rigid-face motion applied over the solid element patch Γ_i. Let $\mathbf{x} \in \Gamma_i$ denote the position of a point on that face, and let $\mathbf{x}_{c,i}$ denote the centroid of the face. The displacement field imposed on the face is written as

$$\mathbf{u}^{(s)}(\mathbf{x}) = \mathbf{t}_{\Gamma i}^{(s)} + \boldsymbol{\theta}_{\Gamma i}^{(s)} \times (\mathbf{x} - \mathbf{x}_{c,i}), \qquad \mathbf{x} \in \Gamma_i,$$

where $\mathbf{t}_{\Gamma i}^{(s)}$ is the translational part of the rigid-face motion and $\boldsymbol{\theta}_{\Gamma i}^{(s)}$ is its rotational part. The corresponding 6-component solid-side interface vector is

$$\mathbf{q}_{\Gamma i}^{(s)} = \begin{bmatrix} \mathbf{t}_{\Gamma i}^{(s)} \\ \boldsymbol{\theta}_{\Gamma i}^{(s)} \end{bmatrix}.$$

Figure 3.6 illustrates this rigid-face kinematic description.

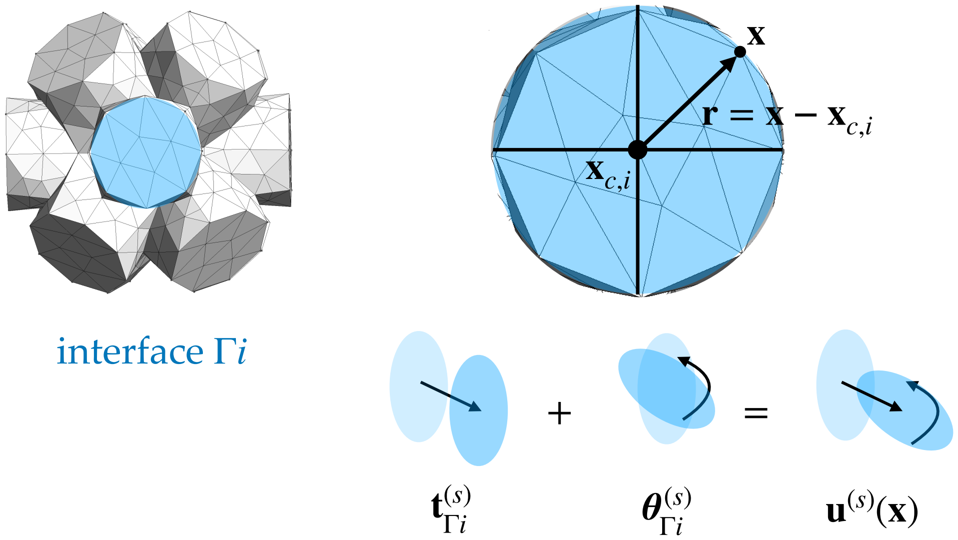


Figure 3.6: Rigid-face kinematics used on a solid interface patch Γ_i. The solid displacement field on the face is restricted to the form $\mathbf{u}^{(s)}(\mathbf{x}) = \mathbf{t}^{(s)}_{\Gamma i} + \boldsymbol{\theta}^{(s)}_{\Gamma i} \times (\mathbf{x} - \mathbf{x}_{c,i})$, where $\mathbf{x}_{c,i}$ is the face centroid and $\mathbf{x}$ is a point on the interface.

This is a kinematic rigid-face coupling: the solid face is restricted to a rigid motion compatible with the six beam-end degrees of freedom. Higher-order face warping and non-rigid in-plane deformation are discarded in order to keep the interface low-dimensional while preserving a clean exchange format between the solid and beam submodels.

Once the solid subproblem has been solved, nodal reactions are summed over the face to obtain a resultant force and resultant moment,

$$\mathbf{F}^{(s)}_{\Gamma i} = \sum_{a \in \Gamma_i} \mathbf{f}_a, \qquad \mathbf{M}^{(s)}_{\Gamma i} = \sum_{a \in \Gamma_i} (\mathbf{x}_a - \mathbf{x}_{c,i}) \times \mathbf{f}_a.$$

These are assembled into the 6-component solid-side interface resultant

$$\mathbf{f}^{(s)}_{\Gamma i} = \begin{bmatrix} \mathbf{F}^{(s)}_{\Gamma i} \\ \mathbf{M}^{(s)}_{\Gamma i} \end{bmatrix}.$$

analogously to the beam submodels.

A number of alternative beam–solid interface coupling strategies could in principle be used in place of the rigid-face approach adopted here. These strategies differ mainly in how strongly compatibility is enforced, whether additional interface unknowns are introduced, and how naturally they accommodate non-matching discretizations. Table 3.4 summarizes representative coupling families and their principal trade-offs and motivation to stick to the rigid-face MPC (multi-point constraint) approach.

In this thesis, the rigid-face MPC strategy is retained primarily because it matches the beam–solid coupling concept used by Portela, Greer *et al.* [12], enabling a more direct comparison between the precomputed reduced-order nodal approach of that work and the fully coupled hybrid formulation developed here. Beyond this methodological consistency, the rigid-face MPC constraint also provides a small and well-conditioned interface system, matches the natural translational and rotational beam-end degrees of freedom, and yields a direct interpretation of interface resultants in terms of forces and moments. The price paid is that detailed face compliance is not transmitted across the interface. This is a limitation of this choice, possibly at the origin of part of the modeling errors.

Table 3.4: Common interface coupling strategies and their trade-offs.

	Advantages	**Disadvantages**
Rigid-face MPC (multi-point constraint) [30, 31] (this work)	Small effective interface when tied to a beam reference node; simple implementation; direct mapping to beam translational and rotational DOFs; robust in legacy finite-element codes.	Constrains the interface patch to beam-like rigid-section kinematics; neglects face warping and higher-order deformation; accuracy depends on the size and location of the constrained patch.
Penalty method [32]	Easy to implement; no additional unknowns; can preserve matrix symmetry when formulated consistently.	Constraint enforcement is approximate; penalty parameter tuning is required; overly large penalties can cause ill-conditioning.
Lagrange multiplier method [33]	Exact constraint enforcement; no penalty tuning; physically interpretable interface forces.	Introduces additional unknowns; leads to saddle-point systems; stability may depend on the choice of multiplier space.
Nitsche's method [34, 35]	Consistent weak enforcement without multiplier DOFs; suitable for non-matching meshes; can preserve variational consistency.	Requires stabilization parameters; formulation and implementation are more complex than direct MPC or penalty coupling.
Mortar method [36]	Well suited for non-matching meshes; accurate and conservative interface transfer; flexible for contact and domain decomposition.	Requires interface projection/integration machinery; implementation is more involved; may introduce additional interface variables depending on formulation.
RBE3 / averaged MPC (Rigid Body Element, type 3 / averaged multi-point constraint) [37]	Distributes forces and moments over an interface patch without imposing fully rigid face motion; reduces local over-stiffening compared with rigid MPCs.	Still an approximate engineering constraint; accuracy depends on weighting and patch selection; not generally derived from a fully variational interface formulation.

3.5.2. Interface Equilibrium and Coupling Conditions

Let Ω_s denote the solid-joint domain and Ω_b the beam-strut domain. Their interaction occurs through the set of interfaces Γ_i. At each interface, the coupled problem is governed by two conditions:

1. *Kinematic compatibility*, which enforces consistency between the beam-end motion and the rigid-face motion imposed on the solid patch.
2. *Interface equilibrium*, which enforces balance of the force and moment resultants exchanged by the two submodels.

also written as

$$\mathbf{q}_{\Gamma i}^{(s)} - \mathbf{q}_{\Gamma i}^{(b)} = \mathbf{0}, \qquad \mathbf{f}_{\Gamma i}^{(s)} + \mathbf{f}_{\Gamma i}^{(b)} = \mathbf{0},$$

where $\mathbf{q}_{\Gamma i}^{(s)}$ and $\mathbf{q}_{\Gamma i}^{(b)}$ are the solid-side and beam-side descriptions of the interface motion, and $\mathbf{f}_{\Gamma i}^{(s)}$ and $\mathbf{f}_{\Gamma i}^{(b)}$ are the associated 6-component interface resultants. Once compatibility is enforced, the common interface motion is denoted by

$$\mathbf{q}_{\Gamma i}^{(s)} = \mathbf{q}_{\Gamma i}^{(b)} = \mathbf{q}_{\Gamma i}.$$

The nonlinear coupling problem solved later in this chapter is an interface equilibrium problem posed on these common interface vectors.

3.5.3. Monolithic and Staggered Coupling Strategies

Hybrid beam–solid modeling can be formulated either as a monolithic problem, where all degrees of freedom are assembled into a single global nonlinear system, or as a partitioned problem where each submodel is solved independently and coupled through an interface iteration. A monolithic formulation offers the strongest algebraic consistency because the solid, beam, and coupling terms are solved simultaneously. However, it requires intrusive modifications to the solver infrastructure and a unified treatment of element kinematics and constitutive updates in implementation. Given the nature of the Summit framework and the different kinematic descriptions of solid and beam elements, such an approach was deemed impractical and too complex on the implementation side for this thesis.

The staggered approach provides an important practical advantage: the beam and solid components remain separate black-box submodels from the perspective of the interface solver. This modularity makes it possible to combine heterogeneous model types, evaluate independent joint and beam solves in parallel, and replace individual joint models in the future by reduced or data-driven alternatives. These properties are particularly relevant for the long-term goal of studying nonlinear joint physics without resolving every strut and joint as one monolithic three-dimensional model.

Conceptually, in a monolithic formulation, the solid and beam subproblems are not solved separately. Instead, their equilibrium equations are assembled into a single global residual system, in which interface equilibrium is enforced directly at the algebraic level. Denoting the full algebraic unknown vectors of the solid and beam subproblems by $\mathbf{d}^{(s)}$ and $\mathbf{d}^{(b)}$, and the interface resultants contributed by the solid and beam sides by $\mathbf{f}_\Gamma^{(s)}$ and $\mathbf{f}_\Gamma^{(b)}$, the coupling condition is

$$\mathbf{f}_\Gamma^{(s)} + \mathbf{f}_\Gamma^{(b)} = \mathbf{0}.$$

This condition is not imposed as an outer iteration, but is embedded directly into the global residual through the coupled solid and beam equilibrium equations. At the subdomain level, one may write

$$\mathbf{R}_\mathrm{s}^\mathrm{int}(\mathbf{d}^{(s)}) - \mathbf{R}_\mathrm{s}^\mathrm{ext} + \mathbf{C}_\mathrm{s}^T \mathbf{f}_\Gamma^{(s)} = \mathbf{0},$$

$$\mathbf{R}_\mathrm{b}^\mathrm{int}(\mathbf{d}^{(b)}) - \mathbf{R}_\mathrm{b}^\mathrm{ext} + \mathbf{C}_\mathrm{b}^T \mathbf{f}_\Gamma^{(b)} = \mathbf{0},$$

where $\mathbf{C}_\mathrm{s}$ and $\mathbf{C}_\mathrm{b}$ denote the operators that distribute the interface actions to the solid and beam degrees of freedom. Since interface equilibrium requires $\mathbf{f}_\Gamma^{(s)} = -\mathbf{f}_\Gamma^{(b)}$, the coupled problem can be assembled into one nonlinear residual

$$\mathbf{R}_\mathrm{mono}(\mathbf{d}) = \begin{bmatrix} \mathbf{R}_\mathrm{s}(\mathbf{d}^{(s)}, \mathbf{d}^{(b)}) \\ \mathbf{R}_\mathrm{b}(\mathbf{d}^{(b)}, \mathbf{d}^{(s)}) \end{bmatrix} = \mathbf{0}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}^{(s)} \\ \mathbf{d}^{(b)} \end{bmatrix},$$

where the dependence of $\mathbf{R}_\mathrm{s}$ on $\mathbf{d}^{(b)}$ and of $\mathbf{R}_\mathrm{b}$ on $\mathbf{d}^{(s)}$ arises through the interface coupling terms.

Linearization in a Newton–Raphson scheme then gives the coupled tangent system

$$\begin{bmatrix} \mathbf{K}_\mathrm{ss} & \mathbf{K}_\mathrm{sb} \\ \mathbf{K}_\mathrm{bs} & \mathbf{K}_\mathrm{bb} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{d}^{(s)} \\ \Delta\mathbf{d}^{(b)} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_\mathrm{s} \\ \mathbf{R}_\mathrm{b} \end{bmatrix},$$

where $\mathbf{K}_\mathrm{ss}$ contains the solid tangent stiffness, $\mathbf{K}_\mathrm{bb}$ the beam tangent stiffness, and $\mathbf{K}_\mathrm{sb}, \mathbf{K}_\mathrm{bs}$ the coupling sensitivities arising from the interface terms. This approach is typically the most robust for strong coupling and highly nonlinear material behavior, provided that consistent tangent operators are available.

The Schur complement offers a useful view: eliminating $\Delta\mathbf{d}^{(s)}$ gives

$$\left(\mathbf{K}_\mathrm{bb} - \mathbf{K}_\mathrm{bs}\mathbf{K}_\mathrm{ss}^{-1}\mathbf{K}_\mathrm{sb}\right) \Delta\mathbf{d}^{(b)} = -\mathbf{R}_\mathrm{b} + \mathbf{K}_\mathrm{bs}\mathbf{K}_\mathrm{ss}^{-1}\mathbf{R}_\mathrm{s},$$

which shows that monolithic coupling implicitly accounts for the solid joint compliance as a (generally dense) correction to the beam-level tangent. This is advantageous for accuracy and convergence, but it can increase implementation complexity and may impact solver scalability in preconditionning and task automation.

In the partitioned, or staggered approach, which is the one adopted in this work, the solid and beam submodels are solved independently and coupled through an interface-level iteration. The interface degrees of freedom are collected in a reduced vector $\mathbf{q}$ obtained by stacking the interface states $\mathbf{q}_{\Gamma i}$, and the coupled problem is recast as the nonlinear interface equilibrium condition

$$\mathbf{R}(\mathbf{q}) = \mathbf{f}^{(s)}(\mathbf{q}) + \mathbf{f}^{(b)}(\mathbf{q}) = \mathbf{0}.$$

For a given interface state $\mathbf{q}^{(k)}$, the solid and beam solvers are executed independently to compute interface reactions, and the residual $\mathbf{R}(\mathbf{q}^{(k)})$ is evaluated. An updated interface state is then obtained through an iterative correction,

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \Delta\mathbf{q}^{(k)},$$

where the correction $\Delta\mathbf{q}^{(k)}$ is computed using one of the numerical strategies described in section 3.6.

In summary, a monolithic strategy maximizes coupled consistency and is attractive when robust tangent operators are available for all element and interface contributions. Its main drawback is implementation complexity. The partitioned strategy adopted here sacrifices some monolithic consistency but preserves modularity: solid joints, beam struts, and future surrogate joint models can be treated as separate operators coupled only through interface kinematics and resultants. This modularity is central to the method developed in this thesis.

Demonstrating this approach is relevant for larger high-fidelity metamaterial simulations, particularly in cases where local instability, post-buckling deformation, or low strut aspect-ratio make pure beam-element models insufficient. The hybrid formulation therefore targets applications in which joint-related parameters must be modeled more accurately than in a beam-only idealization, while still avoiding the cost of a fully solid lattice model. This is especially relevant for stretching-dominated failure studies, where localized instability and boundary-condition sensitivity can strongly influence the observed response, and beam-only models are only sufficiently exact for linear elastic behavior modeling, as discussed in chapter 2.

3.6. Implementation of the Hybrid Element Staggered Model

The objective of the hybrid coupling algorithm is to determine interface motions such that all solid-joint and beam-strut submodels satisfy equilibrium at every coupling interface. The solid and beam models are solved independently using the existing Summit solver infrastructure and are treated as black-box operators that map prescribed interface kinematics to interface reaction resultants. The coupling problem is therefore expressed and solved entirely at the interface level. This section introduces the global interface formulation and the implemented coupling loop. The numerical solution methods used to update the interface state are discussed in detail in chapter 4.

The solver implementation is designed primarily for robust convergence across the models tested in this thesis, with reasonable scaling as the number of interfaces increases. Detailed optimization of convergence speed, iteration count, and solver parameter tuning is outside the main scope of this work.

3.6.1. Global Interface Unknown Vector

The first step is the construction of the global interface unknown vector. Let N_Γ denote the number of coupled interfaces, that is, the number of beam ends attached to solid faces in the network. For each interface Γ_i, the rigid-face coupling strategy of subsection 3.5.1 defines the six-component interface state

$$\mathbf{q}_{\Gamma i} = \begin{bmatrix} u_x & u_y & u_z & \theta_x & \theta_y & \theta_z \end{bmatrix}_{\Gamma i}^{\mathsf{T}} \in \mathbb{R}^6,$$

where (u_x, u_y, u_z) are translations and $(\theta_x, \theta_y, \theta_z)$ are rotation-vector components. The global interface vector is assembled by stacking all interface states,

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{\Gamma 1} \\ \mathbf{q}_{\Gamma 2} \\ \vdots \\ \mathbf{q}_{\Gamma N_\Gamma} \end{bmatrix} \in \mathbb{R}^n, \qquad n = 6N_\Gamma.$$

For each joint model, the classified faces fall into three mutually exclusive categories. First, *coupled faces* are those listed in the beam–joint connectivity. Each coupled face defines one coupling interface Γ_i and therefore contributes one six-component interface state $\mathbf{q}_{\Gamma i}$ to the global interface vector $\mathbf{q}$. Second, *prescribed load-case faces* are those listed in the load case but not in the connectivity. These faces are used to apply supports or imposed rigid-face motions through ramp-scaled Dirichlet data and do not contribute unknowns to $\mathbf{q}$. Third, any remaining classified faces are neither coupled nor prescribed and therefore remain free, acting as natural Neumann boundaries in the joint solve. This categorization is illustrated in Figure 3.7.

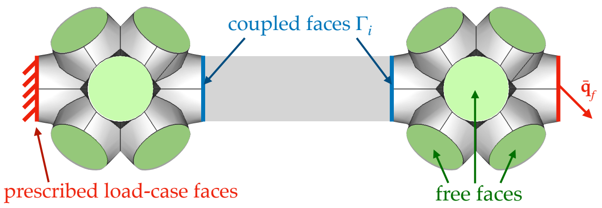


Figure 3.7: Classification of solid faces in the hybrid formulation. Blue faces are coupled faces, each associated with a beam end and an interface state $\mathbf{q}_{\Gamma i}$. Red faces are prescribed load-case faces, on which Dirichlet data such as fixed supports or imposed rigid-face motions $\bar{\mathbf{q}}_f$ may be applied. Green faces are free faces and therefore act as natural Neumann boundaries in the joint solve.

With this classification established, for each joint solve a face motion vector $\mathbf{q}_{\text{faces}} \in \mathbb{R}^{6N_f}$ is assembled in three steps. First, the ramp-scaled prescribed values are written to the prescribed load-case faces. Second, the coupled faces are overwritten by the current interface states $\mathbf{q}_{\Gamma i}$. Third, the remaining classified faces are left free and therefore act as natural Neumann boundaries. The face vector uses the ordering $[T_x, T_y, T_z, R_x, R_y, R_z]$ per face, so rigid rotations are prescribed through the face kinematics even though the solid model itself carries only translational nodal degrees of freedom.

Boundary conditions on the joint model are enforced through a per-face translation mask. During each ramp step, the code uses a two-stage masking strategy on load-case faces: at iteration 0 all three translations are clamped to anchor the configuration, while for later iterations only the prescribed translation components remain clamped, for example fixed z with free x and y. This avoids over-constraining tangential motion during the interface iterations while keeping the independently solved submodels properly anchored at the start of each load increment. In the present implementation, coupled faces do not simultaneously carry additional load-case entries. If interface Neumann loading is to be added in future, the natural insertion point is the interface residual assembly discussed in the next subsection.

3.6.2. Interface Reactions and Global Residual

For a given $\mathbf{q}$, each solid model is solved with Dirichlet boundary conditions assembled from: (i) prescribed external boundary conditions from the load case, and (ii) the current interface states $\mathbf{q}_{\Gamma i}$ for faces coupled to beams. The solid solve returns nodal reactions, which are integrated on each coupled face to yield a six-component resultant vector,

$$\mathbf{f}^{(s)}_{\Gamma i}(\mathbf{q}) = \begin{bmatrix} F_x & F_y & F_z & M_x & M_y & M_z \end{bmatrix}^{\mathsf{T}}_{\Gamma i}.$$

Similarly, each beam model is solved with Dirichlet end conditions extracted from the corresponding displacement interface states at its two ends. The beam solve returns end reactions (forces and moments) mapped consistently to the same six-component space:

$$\mathbf{f}^{(b)}_{\Gamma i}(\mathbf{q}).$$

The interface residual for Γ_i is defined as the equilibrium mismatch

$$\mathbf{r}_{\Gamma i}(\mathbf{q}) = \mathbf{f}^{(s)}_{\Gamma i}(\mathbf{q}) + \mathbf{f}^{(b)}_{\Gamma i}(\mathbf{q}) \in \mathbb{R}^6.$$

Neumann boundary conditions on selected interfaces can be incorporated in the same interface-level formulation by adding a prescribed resultant $\mathbf{f}_{\Gamma i}^{(N)}$ to the residual. For those interfaces, the equilibrium condition becomes

$$\mathbf{r}_{\Gamma i} = \mathbf{f}_{\Gamma i}^{(s)} + \mathbf{f}_{\Gamma i}^{(b)} - \mathbf{f}_{\Gamma i}^{(N)},$$

that is, a direct offset of the residual entries associated with the affected interfaces, scaled by λ_k when load stepping is used. This was not implemented in the current codebase, but it is a relatively straightforward extension because the residual is already assembled interface by interface.

The global interface residual is assembled by stacking all interface residuals,

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} \mathbf{r}_{\Gamma 1}(\mathbf{q}) \\ \vdots \\ \mathbf{r}_{\Gamma N_\Gamma}(\mathbf{q}) \end{bmatrix} \in \mathbb{R}^n.$$

The coupled hybrid equilibrium problem is therefore the nonlinear system

$$\boxed{\mathbf{R}(\mathbf{q}) = \mathbf{0}.}$$

3.6.3. Outer Load Stepping and Inner Coupling Iterations

The network solve is performed incrementally using a scalar ramp parameter $\lambda \in [0, 1]$ that scales applied loads or prescribed kinematics. Let λ_k denote the ramp value at load step k. For each λ_k, an inner interface-equilibrium iteration is executed until convergence or until the maximum number of coupling iterations is reached. The overall structure of this algorithm is summarized in the flowchart in Figure 3.8.

For a fixed interface state $\mathbf{q}^{(m)}$, the expensive part of one coupling iteration is the evaluation of all submodels. This phase is detailed in the flowchart in Figure 3.9. First, iteration data are prepared from the current interface state together with the ramp-scaled prescribed values. The joint and beam branches are then evaluated independently. On the joint side, the face motion vectors $\mathbf{q}_{\text{faces}}^{(j)}$ are assembled, the face masks are applied, and each solid joint model returns its interface resultants $\mathbf{f}_{\Gamma i}^{(s)}$. On the beam side, the end states are extracted from the same interface vector $\mathbf{q}^{(m)}$, and each beam model returns its end resultants $\mathbf{f}_{\Gamma i}^{(b)}$. Only after these independent subproblems have been solved are the resultants returned to the global residual assembly.

This evaluation phase is also the natural location for parallelization in the hybrid staggered model. Algorithmically, all joint subproblems and all beam subproblems are independent for fixed interface kinematics and may therefore be evaluated concurrently. In the present implementation, this property is exploited primarily through the independent joint solves, which dominate the computational cost, while the beam branch remains comparatively lightweight. The staggered coupling itself, however, remains serial at the outer iteration level: residual assembly, convergence checks, and interface-state updates are carried out only after the submodel evaluations for the current iterate have been completed.

The masking logic applied on the joint side is part of this same evaluation phase. At iteration $m = 0$ of each load step, all three translations on the prescribed load-case faces are clamped to anchor the independently solved joint models. For later iterations, only the prescribed translational components remain clamped. Coupled faces are overwritten by the current interface states $\mathbf{q}_{\Gamma i}^{(m)}$, while the remaining faces act as natural Neumann boundaries. Once both solid and beam resultants have been recovered, the interface residuals $\mathbf{r}_{\Gamma i}$ are assembled, the global norm $\|\mathbf{R}\|_2$ is checked against the tolerance, and a new interface correction $\Delta\mathbf{q}^{(m)}$ is computed when required.

Why incremental ramping is kept even with Newton-type methods.
Although the Newton or JFNK (Jacobian-free Newton-Krylov) methods introduced in the following chapter 4 can converge in few iterations when started sufficiently close to the solution, ramping improves robustness under strong geometric nonlinearity and becomes essential in a conceptual material

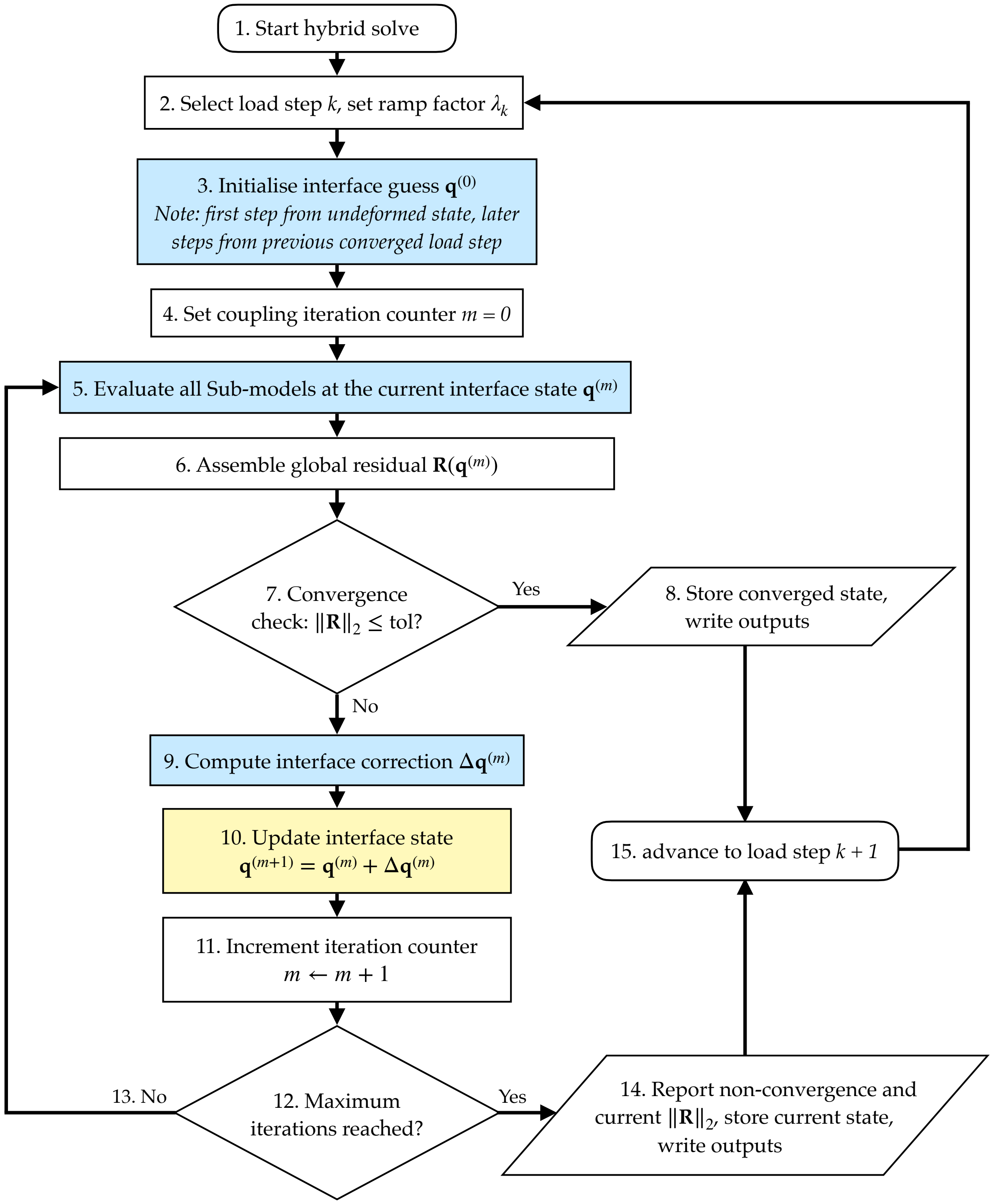


Figure 3.8: Overall flowchart of the hybrid coupling algorithm. For each load step k, a ramp factor λ_k is applied and an inner interface-equilibrium iteration is performed until convergence or until the maximum number of iterations is reached. The light blue boxes highlight the main interface-level operations, while the yellow box marks the interface-state update step.

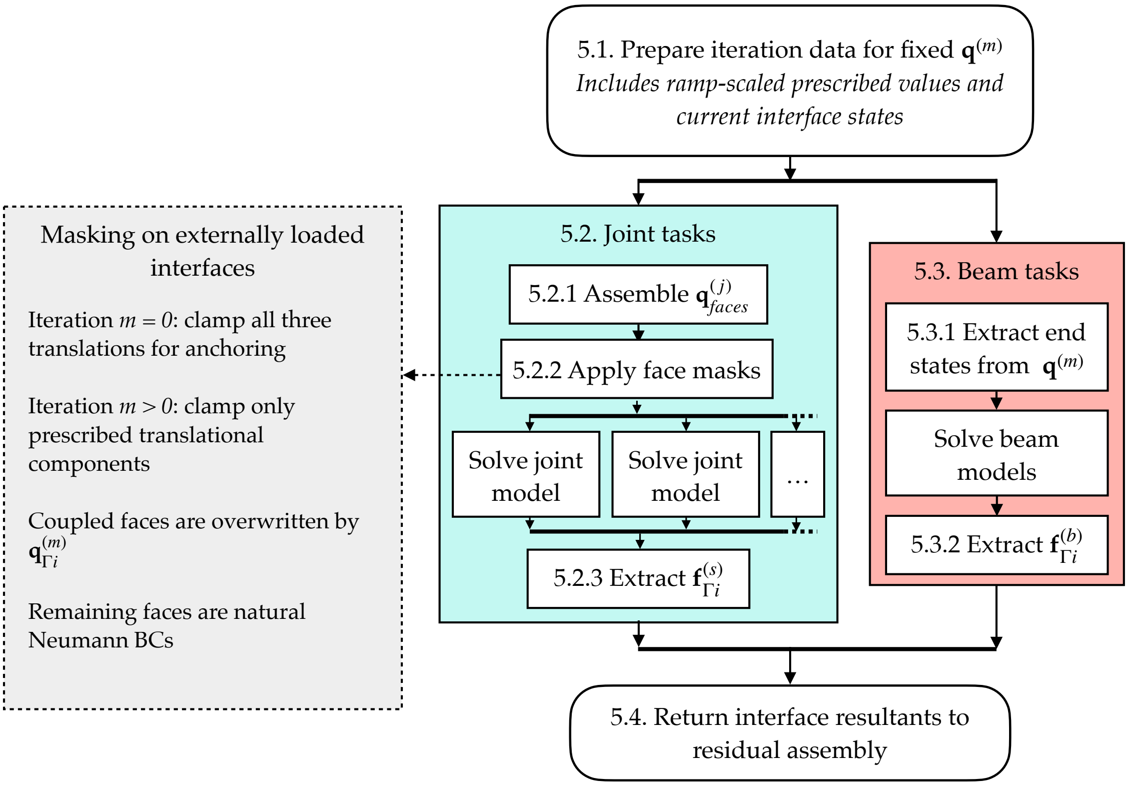


Figure 3.9: Detailed view of Step 5 from Figure 3.8: evaluation of all submodels for a fixed interface state $\mathbf{q}^{(m)}$. The turquoise and red blocks distinguish the joint and beam evaluation branches. Since these branches are independent for fixed $\mathbf{q}^{(m)}$, they define the natural parallelization point of the hybrid algorithm.

plasticity introduction. Plasticity is path-dependent, and equilibrium must be enforced at each increment to maintain consistent internal variables (plastic strains, yield status). In that regime, the coupling algorithm operates inside each load increment, using the converged state at λ_k as the initial guess for all inner iterations of the loadstep λ_{k+1}.

The inclusion of incremental ramping also provides a natural way to generate stress–strain-style curves with multiple converged data points. In this way, interface equilibrium is enforced at every reported load increment. Adaptive ramp sizing could further improve efficiency by using smaller increments in strongly nonlinear regimes and larger increments in nearly linear regimes. This was not implemented in the present work, but is a useful direction for future performance improvement.

4

Nonlinear Solution Methods for Staggered Hybrid Coupling

The hybrid beam-3D continuum coupling problem leads to a nonlinear interface equilibrium system in which residual evaluations are provided by black-box beam and solid solvers. Achieving robust convergence under strong nonlinearity, while keeping the number of residual evaluations low, requires a careful balance between globally convergent methods and fast local accelerators. This chapter collects the numerical tools used to strike that balance and motivates the algorithmic choices made in this work.

This chapter presents the numerical algorithms used to solve the staggered nonlinear hybrid coupling problem introduced in chapter 3. Starting with section 4.1, a case is made for the approach necessary to solve a problem of the nature of this interface problem. Section 4.2 introduces baseline fixed-point iteration and Anderson acceleration methods, considered in the thesis but ultimately underperforming compared to other approaches. Section 4.3 describes the scaling and preconditioning strategies used across solvers to improve convergence. Section 4.4 details the preconditioned residual descent method, while section 4.5 covers Newton-based methods, including full-Jacobian and Jacobian-free Newton–Krylov approaches. The key methodology section of this chapter is section 4.6. It details and motivates the two-stage preconditioned residual descent to Jacobian free Newton-Krylov (PRD–JFNK) switching strategy, the final solver design used throughout the large metamaterial simulations in this report. Table 4.1 presents all solvers considered and used in this thesis and is a useful summary of all previous sections of the chapter. Finally, section 4.7 provides computational cost models based on the influence of different model parameters, building up expected scaling trends further motivated by small-scale experimental data.

4.1. Nature of the Nonlinear Interface Problem and Solver Design Strategy

The nature of the nonlinear problem of the staggered approach is not a conventional monolithic finite element equilibrium solve. In a monolithic formulation, the global displacement vector, internal forces, tangent stiffness matrix, and boundary conditions all belong to one assembled system. In the staggered formulation used in this thesis, the unknowns solved by the coupling algorithm are instead the interface translations and rotations that connect beam struts to three-dimensional solid joint regions. For a model with N_Γ beam–solid interfaces, the coupling vector is

$$\mathbf{q}_\Gamma \in \mathbb{R}^{6N_\Gamma},$$

where each interface contributes three translational and three rotational degrees of freedom.

For a given trial interface state $\mathbf{q}_\Gamma$, the beam and solid submodels are solved separately. The coupling residual is then constructed from the mismatch of the interface reactions predicted by the two sides of each coupling boundary. The nonlinear interface problem can therefore be written abstractly as

$$\mathbf{R}(\mathbf{q}_\Gamma) = \mathbf{0}, \qquad \mathbf{R} : \mathbb{R}^{6N_\Gamma} \to \mathbb{R}^{6N_\Gamma}.$$

The residual vector contains force and moment equilibrium components, while the unknown vector contains translations and rotations. This mixture of physical units makes the raw Euclidean norm of the residual a poor measure unless the problem is scaled (as done for the convergence metric used in the implementation) or preconditioned.

A residual evaluation is expensive. Each call to $\mathbf{R}(\mathbf{q}_\Gamma)$ requires applying the current interface state to the beam and solid submodels, solving those submodels, extracting the resulting interface reactions, and assembling the imbalance into a global residual vector. The coupling solver therefore operates on a black-box nonlinear residual: the residual can be evaluated, but the full interface Jacobian

$$\mathbf{J}(\mathbf{q}_\Gamma) = \frac{\partial \mathbf{R}}{\partial \mathbf{q}_\Gamma}$$

is not naturally available from one assembled global tangent matrix.

This black-box structure strongly influences the solver design. A full Newton method is attractive because it can provide fast local convergence, but forming $\mathbf{J}$ by finite differences requires many residual evaluations. Since the interface dimension is $6N_\Gamma$, a dense finite-difference Jacobian would require at least $6N_\Gamma + 1$ residual evaluations per rebuild if each degree of freedom is perturbed separately, and even more considering the complex coupling of all interfaces through lattice connectivity. This cost becomes prohibitive as the number of interfaces grows. On the other hand, methods that require only residual evaluations, such as fixed-point iterations or residual descent, are cheaper per iteration but may converge slowly or stagnate in strongly coupled nonlinear cases.

The interface problem is also scale-heterogeneous. Translational interface degrees of freedom are measured in units of length, while rotational degrees of freedom are dimensionless angles. Their conjugate residual components are forces and moments, respectively. These quantities can differ by orders of magnitude and do not have the same mechanical sensitivity. Without scaling, a solver may reduce one part of the residual while neglecting another, or choose step lengths that are appropriate for translations but inappropriate for rotations. This is why preconditioning is also a necessary unit-balancing operation.

The nonlinearities encountered in the models ran further complicate the interface solve. The lattice models can undergo large rotations, geometric softening, snap-through-like behavior, and post-buckling deformation. In such regimes, the local linearization used by Newton or Krylov methods can be unreliable if the current iteration step is still far from equilibrium. On the other hand, purely residual-based fixed-point methods may be robust initially but inefficient near convergence. This creates a division of solver roles. Firtsly, a global residual-reduction method is useful for reaching a reasonable neighbourhood of equilibrium, then, a Newton–Krylov method is useful once the residual is sufficiently smooth and informative.

The solver strategy developed in this chapter follows directly from these properties. Fixed-point and Anderson-type methods are first considered because they require only residual evaluations and are useful baselines that can converge almost-linear small problem sizes. Preconditioned residual descent is then introduced as a more robust global residual-reduction method. Newton and Jacobian-free Newton–Krylov methods are considered for faster local convergence, with the Jacobian action approximated by finite differences of the black-box residual. The final solver design in section 4.6 reflects these points: PRD is used to obtain robust initial progress within each load step, and JFNK is used as a local accelerator once the interface state is suitable for Newton-type correction.

4.2. Baseline Fixed-Point and Anderson-Type Methods

The simplest class of nonlinear solvers considered in this work uses only residual evaluations of the interface equilibrium problem. For these methods, no global tangent matrix is required, and the beam and solid submodels only need to provide interface reactions for a prescribed interface state. They provide useful baseline algorithms and motivate the scaling and preconditioning strategies used by the final solver.

However, residual-only methods must be treated with caution. The interface vector contains both translations and rotations, while the residual contains both forces and moments. A direct update based on the unscaled residual is therefore not meaningful unless the residual components are first balanced by a physically motivated scaling or approximate inverse Jacobian action. For this reason, the fixed-point and Anderson-type methods below are written in terms of a preconditioned residual correction.

4.2.1. Fixed-Point Iteration with Aitken Relaxation

A basic residual-correction iteration updates the interface state according to

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} - \omega^{(k)}\mathbf{P}\mathbf{R}(\mathbf{q}^{(k)}), \tag{4.1}$$

where $\mathbf{R}(\mathbf{q}^{(k)})$ is the current interface residual, $\mathbf{P}$ is a scaling or approximate inverse Jacobian operator, and $\omega^{(k)}$ is a relaxation factor, with k the current iteration step. $\omega^{(k)}$controls "how much" of the correction is applied. The product $\mathbf{P}\mathbf{R}(\mathbf{q}^{(k)})$ represents the correction direction predicted from the current residual, while the minus sign means the update is taken in the direction that reduces this residual.

In this form, the method is still a fixed-point iteration because the next state $\mathbf{q}^{(k+1)}$ is computed directly from the current state $\mathbf{q}^{(k)}$, without solving a linearized Newton system.

For conservative values of $\omega^{(k)}$, the method can provide robust initial progress, even robust convergence in simple small-scale problems. The main limitation is slow asymptotic convergence, especially when the interface coupling is strong or when the effective fixed-point map is less accurate in a specific problem. This behavior is common in partitioned coupling algorithms, where the subproblems are solved separately and equilibrium is enforced iteratively at the interface. Dynamic relaxation, including Aitken-type relaxation, is therefore often used to improve the behavior of fixed-point coupling schemes [38].

Aitken relaxation updates the scalar relaxation factor using information from the previous iteration of residual corrections. Let

$$\mathbf{z}^{(k)} = \mathbf{P}\mathbf{R}(\mathbf{q}^{(k)})$$

denote the preconditioned residual correction before multiplication by the scalar relaxation factor. The vector $\mathbf{z}^{(k)}$ is therefore the residual after applying the preconditioner or scaling operator. It has the same dimension as the interface state update and can be interpreted as the unrelaxed correction proposed at iteration k. With this definition, the actual update in Equation 4.1 uses $-\omega^{(k)}\mathbf{z}^{(k)}$.

A common dynamic Aitken update [38], written here within the context of the preconditioned residual corrections used in this implementation, is

$$\Delta\mathbf{z}^{(k)} = \mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}, \qquad \omega^{(k)} = -\omega^{(k-1)}\frac{\left(\mathbf{z}^{(k-1)}\right)^{\mathsf{T}}\Delta\mathbf{z}^{(k)}}{\left\|\Delta\mathbf{z}^{(k)}\right\|_2^2},$$

The vector $\Delta\mathbf{z}^{(k)}$ measures how the preconditioned residual correction changed between two consecutive iterations. Aitken relaxation uses this change to estimate whether the previous relaxation factor was too small, too large, or had the wrong sign for the local fixed-point behavior. The numerator is an inner product between the previous correction $\mathbf{z}^{(k-1)}$ and the change in correction $\Delta\mathbf{z}^{(k)}$. It measures how strongly the correction direction is changing relative to the previous step. The notation $\|\cdot\|_2$ is the Euclidean norm.

This correction ensures the resulting value of $\omega^{(k)}$ is an estimate of the relaxation factor that would minimize the residual along the recent sequence of fixed-point corrections. In implementation, $\omega^{(k)}$ is additionally constrained with

$$\omega_{\min} \leq \omega^{(k)} \leq \omega_{\max}.$$

, a reasonable minimum $\omega_{\min}$ and maximum $\omega_{\max}$ cutoff bounds are important in nonlinear problems because the Aitken update alone can otherwise produce aggressive steps if the residual direction changes rapidly (changes sign in some values) between successive iterations.

In this thesis, fixed-point iteration with Aitken relaxation is implemented primarily as a baseline and as a useful conceptual reference. It requires only residual evaluations and minimal memory, but it is not sufficiently reliable as the main nonlinear solver for the strongly nonlinear, large-scale problems of interest. In particular, it leads to stagnation or to relaxation factors that must become very small for stability, rendering the method very slow.

4.2.2. Anderson Acceleration

Anderson acceleration is a residual-only method for accelerating fixed-point iterations. Originally introduced by Anderson [39], it has since become a widely used method for improving the convergence of nonlinear fixed-point methods [40]. Compared with Aitken relaxation, which adapts a single scalar relaxation factor, Anderson acceleration uses a short history of previous iterations and residuals to construct a better update direction.

Again, consider the preconditioned fixed-point form

$$\mathbf{q} = \mathbf{G}(\mathbf{q}), \qquad \mathbf{G}(\mathbf{q}) = \mathbf{q} - \mathbf{z}(\mathbf{q}), \qquad \mathbf{z}(\mathbf{q}) = \mathbf{P}\mathbf{R}(\mathbf{q}).$$

Here, $\mathbf{G}$ denotes one full, unrelaxed fixed-point update. The vector $\mathbf{z}(\mathbf{q})$ is the preconditioned residual correction introduced in the previous subsection, and $\mathbf{P}$ is again a scaling or approximate inverse-Jacobian operator. The corresponding fixed-point residual is

$$\mathbf{F}(\mathbf{q}) = \mathbf{G}(\mathbf{q}) - \mathbf{q} = -\mathbf{z}(\mathbf{q}) = -\mathbf{P}\mathbf{R}(\mathbf{q}).$$

Thus, $\mathbf{F}(\mathbf{q})$ measures how much one full fixed-point step would change the current state. With the sign convention, it is the negative of the preconditioned residual correction. At iteration k, this relation is written as

$$\mathbf{z}^{(k)} = \mathbf{P}\mathbf{R}(\mathbf{q}^{(k)}), \qquad \mathbf{F}^{(k)} = \mathbf{F}(\mathbf{q}^{(k)}) = -\mathbf{z}^{(k)}.$$

The history enters through the changes in the iterations and fixed-point residuals over the most recent iterations. With m being the maximum history length, at iteration k, the number of available history vectors is

$$m_k = \min(m, k),$$

because fewer than m previous changes are available during the first iterations. The stored differences are

$$\Delta\mathbf{q}^{(j)} = \mathbf{q}^{(j)} - \mathbf{q}^{(j-1)}, \qquad \Delta\mathbf{F}^{(j)} = \mathbf{F}^{(j)} - \mathbf{F}^{(j-1)}, \qquad j = k - m_k + 1, \dots, k.$$

Here, $\Delta\mathbf{q}^{(j)}$ measures how the interface state changed between two successive iterations and $\Delta\mathbf{F}^{(j)}$ measures the corresponding change in the fixed-point residual.

These difference vectors are collected in the history matrices

$$\Delta\mathbf{q}_m^{(k)} = \left[\Delta\mathbf{q}^{(k-m_k+1)} \ \cdots \ \Delta\mathbf{q}^{(k)}\right], \qquad \Delta\mathbf{F}_m^{(k)} = \left[\Delta\mathbf{F}^{(k-m_k+1)} \ \cdots \ \Delta\mathbf{F}^{(k)}\right].$$

Each column of $\Delta\mathbf{q}_m^{(k)}$ contains one previous change in the solution vector, and each column of $\Delta\mathbf{F}_m^{(k)}$ contains the corresponding change in the fixed-point residual. Anderson acceleration uses these matrices to estimate how changes at a previous iteration affect changes in the residual over the recent iteration history.

A mixing vector, denoted $\gamma^{(k)}$ is obtained from the (relatively) small least-squares problem

$$\gamma^{(k)} = \arg\min_{\gamma} \left\| \mathbf{F}^{(k)} - \Delta\mathbf{F}_m^{(k)} \gamma \right\|_2 .$$

The vector $\gamma^{(k)}$ contains the coefficients that best approximate the current fixed-point residual using the residual changes stored in the history matrix. This least-squares problem is much smaller than the full nonlinear system because m is small.

The accelerated update is then formed by correcting the current fixed-point step with the same history coefficients,

$$\mathbf{q}^{(k+1)} = \mathbf{G}(\mathbf{q}^{(k)}) - \left(\Delta\mathbf{q}_m^{(k)} + \Delta\mathbf{F}_m^{(k)}\right)\gamma^{(k)}.$$

This update is no longer based only on the latest fixed-point correction. Instead, it combines information from several recent residual changes to reduce the fixed-point residual more effectively. In this sense, Anderson acceleration can be interpreted as a multisecant quasi-Newton-type correction in the subspace spanned by the stored history [40]. The history length m was kept at a length of 5 to 10 iterations in small problems (up to 4 interfaces) and resulted in good behavior.

Anderson acceleration seems like a natural candidate for black-box partitioned coupling problems. It substantially improves over simple fixed-point iteration while avoiding explicit Jacobian assembly. However, it is not automatically robust. Careful tuning of parameter like damping, a restart policy, and scaling are needed in practice for the nonlinear simulations in this thesis. Anderson acceleration is thus an intermediate residual-only accelerator, with as main limitation easy applicability in a range of different problems.

The main conclusion from the fixed-point family is that residual-only methods are attractive but insufficient on their own. They motivate the need for a solver that preserves black-box residual evaluations while adding stronger globalization and local acceleration. This leads first to the preconditioning strategies introduced in section 4.3, then to preconditioned residual descent in section 4.4, and finally to the hybrid PRD–JFNK strategy adopted in section 4.6.

4.3. Scaling and Interface Preconditioning

A said in section 4.2, preconditioning is central to the nonlinear interface solvers used in this thesis. The interface unknowns contain translational and rotational degrees of freedom, while the corresponding residual components contain forces and moments. These quantities have different physical units and can differ by several orders of magnitude. A solver that acts directly on the unscaled residual can therefore choose update directions that are dominated by only one subset of interface components. Preconditioning is a requirement for constructing meaningful residual corrections in this interface residual minimization problem.

Repeating here, the interface equilibrium problem is written as

$$\mathbf{R}(\mathbf{q}) = \mathbf{0}, \qquad \mathbf{R} : \mathbb{R}^n \to \mathbb{R}^n, \qquad n = 6N_\Gamma,$$

where $\mathbf{q}$ collects all interface translations and rotations. Linearizing about a current iteration gives

$$\mathbf{J}(\mathbf{q}^{(k)}) = \frac{\partial \mathbf{R}}{\partial \mathbf{q}}(\mathbf{q}^{(k)}) \in \mathbb{R}^{n\times n},$$

and a pure Newton correction would look like

$$\mathbf{J}(\mathbf{q}^{(k)})\Delta\mathbf{q}^{(k)} = -\mathbf{R}(\mathbf{q}^{(k)}).$$

At large scale, this Jacobian cannot be fully assembled exactly. Instead, the solvers use preconditioners that approximate the dominant stiffness scales of the interface problem.

As stated previously, $\mathbf{P}$ is the preconditioner, an approximate inverse Jacobian operator,

$$\mathbf{P} \approx \mathbf{J}^{-1}.$$

The product $\mathbf{PR}$ has the dimension and interpretation of an interface displacement correction. The preconditioner may be as simple as representative diagonal scaling, or it may be constructed from finite-difference approximations of local interface Jacobian blocks.

Preconditioning enters the implementation in three related ways. In fixed-point and Anderson-type methods, it defines the fixed-point map

$$\mathbf{G}(\mathbf{q}) = \mathbf{q} - \mathbf{P}\mathbf{R}(\mathbf{q}).$$

In PRD (Preconditioned Residual Descent), it defines the residual-correction direction

$$\Delta\mathbf{q}^{(k)} = -\mathbf{P}\mathbf{R}(\mathbf{q}^{(k)}).$$

In right-preconditioned JFNK (Jacobian-free Newton-Krylov), the Newton correction is written as

$$\mathbf{J}(\mathbf{q}^{(k)})\mathbf{P}\mathbf{y} = -\mathbf{R}(\mathbf{q}^{(k)}), \qquad \Delta\mathbf{q}^{(k)} = \mathbf{P}\mathbf{y}.$$

where the Krylov solver operates on the operator $\mathbf{JP}$. If $\mathbf{P}$ is a good approximation of $\mathbf{J}^{-1}$, GMRES requires fewer iterations to solve the linearized Newton system[41], and the overall JFNK method converges faster.

4.3.1. Notation and Role of the Preconditioner

The preconditioners used in this thesis are designed to satisfy two requirements. First, they must balance the physical units and stiffness scales of translations, rotations, forces, and moments. Second, they must be cheap enough to build and apply within a black-box staggered solver. A preconditioner that is too expensive to compute removes the benefit of avoiding a full Jacobian, while a preconditioner that is too simple can lead to slow convergence or GMRES failure.

The preconditioners are therefore organized in increasing levels of complexity. Representative-value diagonal scaling uses analytical stiffness estimates and is almost free to apply. Jacobi diagonal scaling estimates the diagonal of the interface Jacobian numerically. Per-interface block-Jacobi preconditioning captures the local coupling between the six degrees of freedom on one interface. Finally, the block-connectivity-Jacobi preconditioner retains selected off-diagonal couplings implied by the physical beam and joint connectivity of the lattice. Note that these are problem-specific preconditioners.

This hierarchy reflects the computational trade-off of the whole solver. More informative preconditioners require additional residual evaluations to build, but they can reduce the number of PRD iterations and reduce GMRES iteration counts in JFNK. The most appropriate choice for a specific problem depends on the nonlinearity of the load step, the number of interfaces, and the expected cost of one residual evaluation.

4.3.2. Representative-value Diagonal Scaling

The simplest preconditioner exploits the known block structure of each interface. For interface Γ_i, a 6×6 diagonal approximate inverse stiffness is defined as

$$\mathbf{P}_i^{\mathrm{rep}} = \begin{bmatrix} \frac{1}{k_t} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{k_t} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{k_t} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{k_r} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_r} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{k_r} \end{bmatrix},$$

where k_t and k_r are representative translational and rotational stiffnesses. The corresponding global preconditioner is block diagonal,

$$\mathbf{P}^{\mathrm{rep}} = \mathrm{diag}\left(\mathbf{P}_1^{\mathrm{rep}}, \ldots, \mathbf{P}_{N_\Gamma}^{\mathrm{rep}}\right). \tag{4.2}$$

This matrix is not intended to approximate every coupling term in the full Jacobian, but to imitate translational and rotational corrections that have comparable mechanical properties.

The representative stiffnesses are estimated from simple beam-theory scaling laws. For a strut with Young's modulus E, cross-sectional area A, second moment of area I, and reference length ℓ_{ref}, the translational and rotational stiffnesses used are

$$k_t \sim \frac{EA}{\ell_{\mathrm{ref}}}, \qquad k_r \sim \frac{EI}{\ell_{\mathrm{ref}}}.$$

These estimates are simple, but they are inexpensive and already remove the most severe scale imbalance in the residual. The cost of applying $\mathbf{P}^{\mathrm{rep}}$ is negligible.

4.3.3. Jacobian Diagonal Preconditioner

A more problem-specific diagonal preconditioner can be constructed by estimating the diagonal entries of the interface Jacobian using finite differences. For each degree-of-freedom index $j = 1, \dots, n$, let $\mathbf{e}_j$ be the corresponding basis vector. The jth diagonal entry of the Jacobian is approximated as

$$J_{jj} \approx \frac{R_j(\mathbf{q} + \varepsilon \mathbf{e}_j) - R_j(\mathbf{q})}{\varepsilon},$$

where ε is a finite-difference perturbation. The diagonal approximate Jacobian inverse is calculated as

$$\mathbf{P}^{\mathrm{diag}} = \mathrm{diag}\left(\frac{1}{J_{11}}, \dots, \frac{1}{J_{nn}}\right),$$

where additionally, clipping is applied if a diagonal entry is too small or large.

This preconditioner requires one baseline residual evaluation and one perturbed residual evaluation per interface degree of freedom when being computed. Its construction cost is therefore larger than representative-value scaling. The advantage is that it captures the actual stiffness scale of each interface component at the current nonlinear state, which can be important when different interfaces of a metamaterial model experience very different local deformation states.

4.3.4. Per-Interface Block-Jacobian Preconditioner

The diagonal Jacobian preconditioner ignores coupling between translations and rotations on the same interface. A richer approximation is obtained by constructing a 6 × 6 local Jacobian block for each interface. For interface i, $\mathbf{e}_{i,\alpha}$, where $\alpha = 1, \dots, 6$, is a perturbation of the αth local degree of freedom of that interface. The local residual vector on the same interface is written $\mathbf{r}_i$. The local block is approximated by

$$\mathbf{J}_{ii} \approx \begin{bmatrix} \frac{\mathbf{r}_i(\mathbf{q}+\varepsilon \mathbf{e}_{i,1}) - \mathbf{r}_i(\mathbf{q})}{\varepsilon} & \cdots & \frac{\mathbf{r}_i(\mathbf{q}+\varepsilon \mathbf{e}_{i,6}) - \mathbf{r}_i(\mathbf{q})}{\varepsilon} \end{bmatrix} \in \mathbb{R}^{6\times 6}.$$

The corresponding interface-preconditioner block is the inverse of this local Jacobian,

$$\mathbf{P}_i^{\mathrm{block}} \approx \mathbf{J}_{ii}^{-1}.$$

The global preconditioner is again assembled to be block diagonal,

$$\mathbf{P}^{\mathrm{block}} = \mathrm{diag}\left(\mathbf{P}_1^{\mathrm{block}}, \dots, \mathbf{P}_{N_\Gamma}^{\mathrm{block}}\right).$$

This preconditioner captures the coupling between translational and rotational degrees of freedom on the same coupling face. That coupling is mechanically important in bending-dominated deformation, where a force residual can imply both translational and rotational corrections, and a moment residual can imply both rotational and translational changes. The cost of one preconditioner computation is one baseline residual evaluation plus six perturbations per interface. The resulting improvement in nonlinear step quality and GMRES convergence is noticeable. In fact, it is the preconditioner of choice for the pyramidal $r/l = 0.1$ unit cell models ran in chapter 5.

4.3.5. Block-Connectivity-Jacobian Preconditioner

The per-interface block-Jacobi preconditioner still neglects off-diagonal coupling between different interfaces. In the hybrid lattice problem, the most important off-diagonal couplings result from physical connectivity. Two interfaces can be coupled because they are attached to the same solid joint region, or because they lie at opposite ends of the same beam strut. The block-connectivity-Jacobian preconditioner exploits this structure coming from the underlying lattice architecture while still avoiding construction of a full dense interface Jacobian.

For each interface Γ_i, we find the local neighbour set

$$\mathcal{N}_{\Gamma_i} = \{\Gamma_i\} \cup \{\text{interfaces connected to the same solid joint}\} \cup \{\text{interface at the other end of the same beam}\}.$$

For each $\Gamma_j \in \mathcal{N}_{\Gamma_i}$, a 6×6 block $\mathbf{J}_{ij}$ is approximated by perturbing the degrees of freedom of interface j and measuring the residual response on interface i. This produces a sparse block approximation $\tilde{\mathbf{J}}$ of the true interface Jacobian. For example, for a simple three-interface construction, it may have the form

$$\tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22} & \mathbf{J}_{23} \\ \mathbf{0} & \mathbf{J}_{32} & \mathbf{J}_{33} \end{bmatrix},$$

where again, each entry is a 6×6 block.

The preconditioner is computed step by step, for each interface, contributions from already updated neighbouring interfaces are subtracted, and the remaining local correction is computed using the inverse of the resulting $\bar{\mathbf{J}}$.

This preconditioner is tailored to the hybrid beam–solid lattice problem. It is more expensive to build than representative-value or diagonal scaling, but it captures the local connectivity that dominates many difficult nonlinear load steps. In practice, it is particularly useful for cases involving snap-through-like behavior or sudden post-buckling changes, like the octet $r/l = 0.03$ unit cell models ran in chapter 5, where other preconditioners failed to let the more nonlinear loadsteps converge.

4.4. Preconditioned Residual Descent

Preconditioned residual descent (PRD) is used in this work as a global residual-reduction method for the nonlinear interface problem. Its purpose is to produce robust progress when the current interface state is still far from equilibrium. This is especially important at the beginning of a load step, where the residual may be large, sparse, or poorly aligned with the local Newton correction.

Given the current interface state $\mathbf{q}^{(k)}$, PRD forms the correction direction

$$\Delta\mathbf{q}^{(k)} = -\mathbf{P}\mathbf{R}(\mathbf{q}^{(k)}),$$

where $\mathbf{P} \approx \mathbf{J}^{-1}$ is the approximate inverse Jacobian action introduced earlier. The interface state is then updated as

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \alpha^{(k)}\Delta\mathbf{q}^{(k)},$$

where $\alpha^{(k)}$ is a step length.

The residual vector norm, scaled using Equation 4.2, is used as a value for step acceptance. Starting from an initial trial step length, a backtracking line search reduces $\alpha^{(k)}$ until a sufficient residual decrease condition is satisfied,

$$\left\| \mathbf{R}\left(\mathbf{q}^{(k)} + \alpha^{(k)}\Delta\mathbf{q}^{(k)}\right) \right\|_2 \leq \left(1 - c\alpha^{(k)}\right) \|\mathbf{R}(\mathbf{q}^{(k)})\|_2,$$

with $c \in (0, 1)$. This condition plays the same practical role as sufficient-decrease line searches in globalized nonlinear solvers: the step length is reduced until the step acceptance measure decreases sufficiently, as introduced by [42].

To interpret PRD, The method uses the residual as a correction-driving quantity and uses the residual norm as the quantity to be reduced. However, it is not a true steepest-descent method for the objective $\frac{1}{2}||\mathbf{R}(\mathbf{q})||_2^2$, since the exact gradient of that objective is defined as $\mathbf{J}(\mathbf{q})^\mathsf{T}\mathbf{R}(\mathbf{q})$. PRD should therefore be interpreted as a preconditioned residual-correction method rather than as an exact gradient approximation method. The role of the line search is to ensure that the chosen correction still reduces the nonlinear interface imbalance.

The strength of PRD is robustness. Since it only requires residual evaluations and preconditioner multiplications (preconditioner computed once), it is hybrid beam–solid coupling problem by avoiding the need for unnecessary residual evaluations. It can also tolerate imperfect preconditioners, since the line search rejects steps that fail to reduce the residual norm. Its weakness is slow local convergence. Once the iterate is close to equilibrium, residual-correction methods tend to reduce the error only gradually [42], especially when the interface Jacobian has strong off-diagonal coupling or when the residual is poorly scaled, both issues in the metamaterial models.

For this reason, PRD is not used as the only nonlinear solver in the final algorithm. Instead, it serves as a globalization stage by reducing the residual from a poor initial state and producing an interface configuration from which a Newton–Krylov correction becomes more reliable. That is an interface state, where the residuals are shared throughout the residual vector. This division of roles motivates the two-stage PRD–JFNK strategy introduced in section 4.6.

4.5. Newton and Jacobian-free Newton–Krylov Methods

Newton-type methods are attractive because they use local linearization of the nonlinear residual and can converge rapidly once the iterate is sufficiently close to a solution. For the interface equilibrium problem,

$$\mathbf{R}(\mathbf{q}) = \mathbf{0},$$

Newton's method computes a correction by solving

$$\mathbf{J}(\mathbf{q}^{(k)})\Delta\mathbf{q}^{(k)} = -\mathbf{R}(\mathbf{q}^{(k)}), \tag{4.3}$$

followed by the update

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \alpha^{(k)}\Delta\mathbf{q}^{(k)}.$$

Here $\mathbf{J} = \partial\mathbf{R}/\partial\mathbf{q}$ is the interface Jacobian and $\alpha^{(k)}$ is a line-search factor used for globalization.

4.5.1. Full-Jacobian Newton–Raphson Method

The most direct Newton approach is to assemble the interface Jacobian fully by finite differences. For each pair of interface degree of freedom, a pair of perturbations is applied, the residual is re-evaluated, and one column of the Jacobian is approximated. This produces a full numerical Jacobian that can be used in Equation 4.3.

For small interface systems, like a strut with two joints, this method is useful and very effective in practice. It provides a strong local linear model and gives fast convergence when the finite-difference Jacobian is accurate. In this thesis, full-Jacobian Newton solves are therefore useful for small validation problems and for checking the behavior of the interface formulation in the validation experiments of section 5.1.1.

The limitation is cost. For an interface vector of size $n = 6N_\Gamma$, a column-wise finite-difference Jacobian requires at least $n + 1$ residual evaluations per computation, with more residual evaluations in more complex interface systems. Since each residual evaluation involves solving the beam and solid submodels, this cost grows rapidly with the number of interfaces. In addition, storing and solving with the full Jacobian becomes increasingly expensive as the interface system grows. Full-Jacobian Newton is therefore not a feasible method for larger nonlinear lattice simulations.

4.5.2. Jacobian-Free Newton–Krylov Method

Jacobian-free Newton–Krylov methods avoid explicit Jacobian assembly by combining Newton linearization with a Krylov solver for the Newton correction equation [41]. Instead of forming $\mathbf{J}$, the Krylov method only requires products of the form $\mathbf{Jv}$, $\mathbf{v}$ being a Krylov search direction. These products can be approximated by finite differences of the nonlinear residual,

$$\mathbf{J}(\mathbf{q})\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{q} + h\mathbf{v}) - \mathbf{R}(\mathbf{q})}{h}, \tag{4.4}$$

where h is a finite-difference perturbation size.

This formulation is well suited for the hybrid staggered problem. The residual is available as a black-box function of the interface state, but not the full Jacobian. JFNK retains the main advantage of Newton's method, local linearization, while avoiding the construction and storage of the full interface Jacobian.

In practice, the finite-difference perturbation size h is important. If h is too small, subtraction cancellation and solver noise can dominate the directional derivative. If h is too large, truncation error makes the finite-difference approximation inconsistent with the local linearization. In the implementation used here, the base finite-difference scale is adapted using the size of the current iterate and the Krylov vector, together with lower and upper bounds to avoid excessively small or excessively large perturbations. This follows the JFNK requirement that Jacobian–vector products must be accurate enough to support the Krylov solve without requiring explicit Jacobian assembly [41].

GMRES Linear Solve

The Krylov method used in this work is GMRES, the generalized minimal residual method for nonsymmetric linear systems [43]. For a general linear system

$$\mathbf{Ax} = \mathbf{b},$$

GMRES builds the Krylov subspace

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = \operatorname{span}\left\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{m-1}\mathbf{r}_0\right\},$$

where $\mathbf{r}_0$ is the initial linear residual. It then chooses the approximate solution in this subspace that minimizes the linear residual norm. m is the dimension of the Krylov subspace, a larger m gives the solver a larger search space but increases computational memory and time.

In JFNK, the matrix $\mathbf{A}$ is the local Newton Jacobian, and each Krylov matrix–vector product is replaced by the finite-difference operation in Equation 4.4. Restarting is used to bound memory and orthogonalization cost, so the practical performance depends on the restart length, the GMRES tolerance, and the quality of the preconditioner.

Because the Newton correction is solved only approximately by GMRES, the method is best interpreted as an inexact Newton method. The linear-solver tolerance controls how accurately the Newton equation is solved at each nonlinear iteration, and therefore affects both robustness and convergence rate [44]. Solving the linear system too accurately can waste residual evaluations, while solving it too loosely can produce ineffective nonlinear corrections.

Preconditioning in JFNK

Preconditioning is essential for JFNK in the hybrid interface problem. With the convention introduced in section 4.3, $\mathbf{P}$ denotes an approximate inverse Jacobian action. The right-preconditioned Newton correction is therefore written as

$$\mathbf{J}(\mathbf{q}^{(k)})\mathbf{P}\mathbf{y} = -\mathbf{R}(\mathbf{q}^{(k)}), \qquad \Delta\mathbf{q}^{(k)} = \mathbf{P}\mathbf{y}.$$

GMRES is then applied to the operator **JP** rather than to **J** directly.

If **P** captures the dominant inverse stiffness scales of the interface problem, the spectrum of **JP** is more favorable and fewer GMRES iterations are needed. This is particularly important because every Krylov iteration requires at least one additional residual evaluation. Poor preconditioning can therefore make JFNK prohibitively expensive or lead to GMRES failure before a useful Newton correction is obtained.

The same preconditioner hierarchy described in section 4.3 can be reused in JFNK: representative-value scaling, diagonal Jacobian, per-interface block-Jacobian, and block-connectivity-Jacobian. Although the more informative preconditioners are more expensive to construct, but they can reduce the number of Krylov iterations and improve robustness in strongly nonlinear load steps.

4.5.3. Role of Newton-Based Methods in the Implementation

The Newton-based methods occupy two different roles in the implementation. Full-Jacobian Newton is mainly a small-system or diagnostic method. It is useful when the interface dimension is small enough that finite-difference Jacobian construction is affordable. JFNK is the scalable Newton-type method used for larger systems, because it requires only Jacobian–vector products and fully preserves the black-box nature of the residual evaluation.

In the implementation, JFNK is treated as a local accelerator. It assumes that the current iterate is close enough to equilibrium for the finite-difference linearization to be informative. In early iterations of a load step, especially when the interface displacement vector is sparse or the residual is highly localized, the Krylov subspace may not contain useful correction directions. This motivates combining JFNK with PRD: PRD first produces robust global residual reduction, and JFNK is then activated once the interface state is suitable for Newton–Krylov acceleration.

4.6. Final Hybrid PRD–JFNK Solver Design

The final nonlinear solver used in this thesis combines PRD and JFNK. The design follows directly from the strengths and weaknesses identified in previous sections. PRD is robust when the current iteration is far from equilibrium, but it is slow near convergence. JFNK can converge rapidly near equilibrium, but it is sensitive to the quality of the linearization, the finite-difference Jacobian–vector products, and the preconditioner. The hybrid strategy uses PRD as the global stage and JFNK as the local acceleration stage.

At the beginning of each load step, the solver starts with PRD. This stage reduces the residual norm using the preconditioned residual direction and a line search. It also spreads nonzero interface corrections through the coupled lattice, which is important because the initial residual may be concentrated only on the loaded or constrained faces. When sufficient residual reduction or stagnation is detected, the solver switches to JFNK.

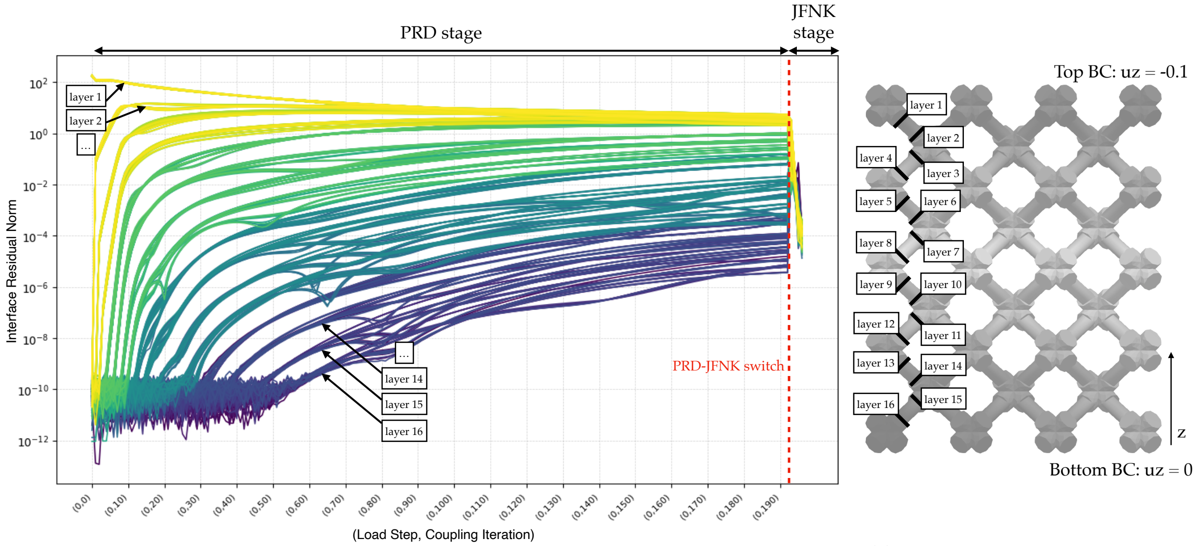
The moment to switch is motivated by two reasons. If PRD is still reducing the residual efficiently, there is no reason to pay the additional cost of Krylov iterations. If PRD begins to stagnate, the residual direction is no longer an efficient correction direction, and a Newton–Krylov correction becomes more attractive. The switching criterion monitors the residual reduction history and activates JFNK when the residual is either sufficiently small or no longer decreasing efficiently under PRD.

This approach is also important for implementation robustness. At iteration zero of a load step, many interface displacement components can be exactly zero, and only a subset of interfaces may experience nonzero residuals. If JFNK is applied immediately, finite-difference Jacobian–vector products can be uninformative, and the Krylov solver may produce poor correction directions. A short PRD stage regularizes this situation by producing a more distributed interface state before the Newton–Krylov linearization is attempted.

To illustrate why the initial PRD stage is useful before activating JFNK, Figure 4.1 shows the evolution of the residual norm at each interface of a $3 \times 3 \times 4$, $r/l = 0.15$ BCC specimen under axial compression with top and bottom faces additionally constrained in x and y-displacement. Each curve represents

$$\|\mathbf{r}_{\Gamma_i}\|_2,$$

where $\mathbf{r}_{\Gamma_i} \in \mathbb{R}^6$ collects the force and moment residual components associated with interface Γ_i. At the beginning of the solve, the residual is concentrated near the loaded top boundary, while many deeper interface layers remain close to the numerical floor. During the PRD stage, this initially localized interface imbalance becomes distributed through progressively deeper layers of the connected model Although the layers are only illustrated on one column of the model, they represent full horizontal layers of interfaces. Once a broader set of interfaces carries non-negligible residual, the JFNK stage can reduce the interface residual norms rapidly across the full specimen.



(a) Evolution of per-interface residual norms across the specimen during coupling iterations

(b) 3x3x4 BCC specimen under axial compression with layer numbering

Figure 4.1: Evolution of per-interface residual norms during the hybrid PRD–JFNK solution of a $3 \times 3 \times 4$ BCC specimen under axial compression. (a) Residual norm $\|\mathbf{r}_{\Gamma_i}\|_2$ at each interface Γ_i as a function of load step and coupling iteration. Interfaces near the loaded top boundary, represented by layers 1–2, become active first, while deeper layers remain close to the numerical floor during the early PRD iterations. As the PRD stage proceeds, residual activity becomes distributed through progressively deeper interface layers. The vertical dashed line marks the switch from PRD to JFNK, after which the interface residual norms are reduced rapidly across the full specimen. (b) Layer numbering used to identify interfaces through the depth of the $3 \times 3 \times 4$ BCC specimen.

The behavior shown in Figure 4.1 supports the role expected from PRD in the final solver design. PRD is not used only to reduce the global residual norm, but also to move the interface state away from a highly localized initial imbalance. This produces a more informative nonlinear state for the subsequent JFNK stage. The figure complements the global residual history in Figure 4.2. This figure shows the rate of convergence during the two stages, while the per-interface residual plot shows how the residual activity becomes distributed through the specimen before Newton–Krylov acceleration becomes effective. Figure 4.2 shows, in one loadstep, that during the initial PRD stage, preconditioned by a representative value diagonal preconditioner, the residual norm decreases slowly but robustly. After the switch, the per-interface block-Jacobian preconditioner is computed, and the JFNK stage rapidly reduces the residual. The dashed line indicates the convergence tolerance. In this case, the tolerance corresponds to a 10^{-6} mismatch in global model external reaction force equilibrium.

The solvers described in this chapter are summarised in Table 4.1. The table is not intended to rank the methods, but to summarise the role each method plays in the present hybrid coupling framework, where robustness from poor initial states, compatibility with black-box residual evaluations, and fast

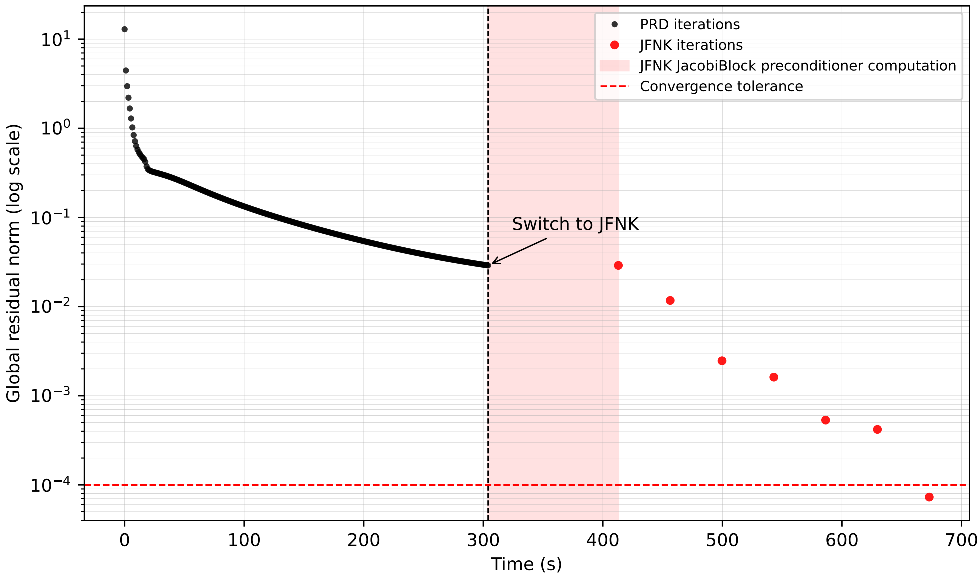


Figure 4.2: Residual history for a nonlinear load step using the hybrid PRD–JFNK solver. PRD provides robust initial residual reduction, while JFNK provides rapid final convergence after block-based preconditioner setup. The dashed line indicates the convergence tolerance.

local convergence are all requirements.

4.7. Computational Cost Model and Expected Scaling

For the purposes of this thesis, the scaling question is how the chosen discretization changes the size and number of nonlinear subproblems that must be solved affect the performance of the approach. The two main modeling questions are therefore, how does solid joint model meshing, size of a metamaterial sample modeled, and connectivity interface count affect computational time and memory?

4.7.1. Fully Solid Element Models

The joint subproblems of the hybrid approach, but also fully solid element lattice models, used as modeling benchmarks are monolithic solid element FE models. Let $N_{\mathrm{dof}}^{\mathrm{sol}}$ be the total number of displacement degrees of freedom in a fully solid lattice model. Let N_{step} be the number of load steps, and let $\bar{N}_{\mathrm{NR}}$ be the average number of Newton iterations per step. The total wall time of a nonlinear quasi-static solid analysis can be approximated as

$$T_{\mathrm{solid}} \approx N_{\mathrm{step}} \bar{N}_{\mathrm{NR}} \left[T_{\mathrm{asm}}\left(N_{\mathrm{dof}}^{\mathrm{sol}}\right) + T_{\mathrm{lin}}\left(N_{\mathrm{dof}}^{\mathrm{sol}}\right) \right],$$

where T_{asm} is the cost of element integration and matrix assembly, and T_{lin} is the cost of one linearized tangent solve.

For standard low-order three-dimensional solid discretizations, T_{asm} and the assembled sparse-matrix storage are both close to linear in $N_{\mathrm{dof}}^{\mathrm{sol}}$[45]:

$$T_{\mathrm{asm}}\left(N_{\mathrm{dof}}^{\mathrm{sol}}\right) = \mathcal{O}\left(N_{\mathrm{dof}}^{\mathrm{sol}}\right), \qquad M_K^{\mathrm{sol}} \approx c_K N_{\mathrm{dof}}^{\mathrm{sol}},$$

where M_K^{sol} is the memory needed to store the assembled tangent matrix and c_K is a storage constant. This is the standard sparse-storage picture described by Davis [45]: the assembled matrix inherits the local connectivity of the finite-element mesh, so the number of stored coefficients remains proportional to the number of unknowns, up to a mesh-dependent constant.

Table 4.1: Roles of the nonlinear solvers considered for the hybrid interface problem.

Method	Role in this thesis	Main strength	Main limitation
Fixed-point with Aitken relaxation	Baseline residual-only coupling method.	Very simple; requires only residual evaluations; low memory.	Slow or stagnant for strongly coupled nonlinear cases; sensitive to scaling.
Anderson acceleration	History-based fixed-point accelerator.	Can accelerate residual-only iterations without explicit Jacobians.	Requires damping/restarts; can be unstable for noisy or strongly nonlinear residuals.
Preconditioned residual descent	Global residual-reduction stage.	Robust from poor initial states; compatible with black-box residuals.	Slow local convergence near equilibrium.
Full-Jacobian Newton–Raphson	Small-system and diagnostic method.	Strong local convergence when a reliable Jacobian is affordable.	Finite-difference Jacobian construction is expensive for many interfaces.
Jacobian-free Newton–Krylov	Local acceleration stage.	Avoids full Jacobian assembly; fast near convergence with good preconditioning.	Sensitive to initial state, finite-difference step size, GMRES tolerance, and preconditioner quality.
Hybrid PRD–JFNK	Final solver design used for difficult nonlinear simulations.	Combines robust global progress with fast local Newton–Krylov convergence.	Requires switching logic and several solver parameters.

The linear solve is typically the dominant term. For sparse direct solvers, the factorization cost is generally superlinear in $N_{\text{dof}}^{\text{sol}}$. This can be written as:

$$T_{\text{lin}}\left(N_{\text{dof}}^{\text{sol}}\right) \approx c_T \left(N_{\text{dof}}^{\text{sol}}\right)^p, \qquad M_{\text{lin}}\left(N_{\text{dof}}^{\text{sol}}\right) \approx c_M \left(N_{\text{dof}}^{\text{sol}}\right)^q,$$

where p is the time-scaling exponent and q is the peak-memory exponent. The reason for introducing these exponents explicitly is that the sparse direct-solver literature does not support a globally linear law in general. George [46] showed for a regular two-dimensional finite-element mesh that nested dissection reduces the factorization cost to $\mathcal{O}(n^3)$ arithmetic operations and $\mathcal{O}(n^2 \log n)$ storage; since such a mesh has $N \sim n^2$ unknowns, these bounds correspond to $\mathcal{O}(N^{3/2})$ work and approximately $\mathcal{O}(N \log N)$ storage. Gilbert and Tarjan [47] extended this result to planar and two-dimensional finite-element graphs, proving $\mathcal{O}(N^{3/2})$ operation counts and $\mathcal{O}(N \log N)$ fill for that class. For more general sparse graphs, Lipton, Rose, and Tarjan [48] developed generalized nested dissection, and Miller, Teng, Thurston, and Vavasis [49] showed that well-shaped d-dimensional finite-element meshes admit separators of size $\mathcal{O}(N^{(d-1)/d})$. Taken together, these results support the standard bulk three-dimensional rule of thumb used in this thesis: representative direct-solver exponents are $p \approx 2$ for factorization time and $q \approx 4/3$ for factorization memory, while effectively lower-dimensional or more slender graphs can exhibit somewhat smaller exponents.

The iterative-solver picture is different. Brandt [50] showed that multilevel methods can achieve $\mathcal{O}(N)$ complexity in the ideal setting, and Stüben [51] reviewed how algebraic multigrid transfers this idea to large unstructured problems. Cleary et al. [52] discuss robustness and scalability of AMG on large unstructured grids, and Griebel, Oeltz, and Schweitzer [53] show that this viewpoint extends to linear elasticity. The appropriate conclusion for the present thesis is therefore a cautious one: iterative solvers can approach linear complexity per Newton step, but only if the preconditioner is strong enough that the iteration count does not itself grow severely with problem size. This caveat matters for slender and low-bulk geometries in particular, because Mishra and Suresh [54] report that iterative performance for thin three-dimensional structures can degrade a lot with increasing aspect ratio.

The corresponding total memory may therefore be written as

$$M_{\text{solid}} \approx M_K^{\text{sol}} + M_{\text{lin}}\left(N_{\text{dof}}^{\text{sol}}\right).$$

The important consequence is that, in a fully solid model, any increase in the global solid DOF count affects every Newton iteration and every load step.

4.7.2. Savings from Reducing the Total Solid DOF Count

Starting with the definition of a reduced-DOF ratio

$$\alpha = \frac{N_{\text{dof}}^{\text{hyb,mono}}}{N_{\text{dof}}^{\text{sol}}},$$

where $N_{\text{dof}}^{\text{hyb,mono}}$ is the total number of degrees of freedom that a hypothetical monolithic hybrid-equivalent discretization would contain, and $N_{\text{dof}}^{\text{sol}}$ is the total DOF count of the fully solid reference model. If $\alpha < 1$, then even before any parallelisation benefit is considered, the reduced DOF count already implies a lower monolithic cost.

Under the direct-solver model above, the corresponding time and memory ratios are typically approximated in the following literature as

$$\frac{T_{\text{lin}}^{\text{hyb,mono}}}{T_{\text{lin}}^{\text{sol}}} \approx \alpha^p, \qquad \frac{M_{\text{lin}}^{\text{hyb,mono}}}{M_{\text{lin}}^{\text{sol}}} \approx \alpha^q.$$

The exponents p and q are intended to represent the superlinear direct-solver behavior discussed above, namely the nested-dissection-based scaling picture established for two-dimensional finite-element graphs by George [46] and Gilbert and Tarjan [47], and then extended to well-shaped higher-dimensional finite-element meshes from results of Miller et al. [49]. In other words, the monolithic comparison developed in this subsection should be read as a model-based estimate that is consistent with the sparse direct-solver literature.

As a concrete example, if the hybrid-equivalent model has only 30% of the DOFs of the solid reference, then $\alpha = 0.3$. In that case the above-mentionned literature predicts

$$\alpha = 0.3, \qquad \alpha^{3/2} \approx 0.16, \qquad \alpha^2 = 0.09, \qquad \alpha^{4/3} \approx 0.20.$$

A 30%-DOF model would still require about 30% of the assembled-matrix storage, but only about 20% of the direct-solver memory under a $q = 4/3$ memory law, and roughly 9% to 16% of the direct-solver time under $p = 3$ to $2, 2$. These numbers are interpreted as order-of-magnitude estimates consistent with the direct-solver scaling discussed above, rather than as exact solver benchmarks. When the monolithic solid solve is superlinear, reducing the global solid DOF count yields disproportionately large savings in both wall time and peak memory.

A second effect is also important. Supposing the solid problem with a total of N solid DOFs is decomposed into k independent subproblems of equal size N/k, and supposing the local solve cost is $T(N) = cN^p$ with $p > 1$. Then the total serial cost becomes

$$k\,c\left(\frac{N}{k}\right)^p = \frac{cN^p}{k^{p-1}},$$

which is smaller than the cost of one monolithic solve. This observation follows directly from the superlinear law $T(N) = cN^p$ assumption. It is relevant because the sparse-direct-solver references above suggest $p > 1$ for large finite-element solid problems [46–49]. Even if the total solid DOF count was unchanged, many smaller solid solves can still be cheaper than one large solve whenever the local linear solver scales superlinearly. The hybrid model combines both effects: the struts are reduced from 3D solids to beam models, which lowers the total DOF count, and the remaining 3D solves are localized to separate joint problems.

4.7.3. Partitioned Hybrid Beam–3D Continuum Model

In the hybrid formulation used here, the joints are represented by local three-dimensional solid submodels, while the struts are represented by beam submodels. Let N_J be the number of joint submodels, let N_B be the number of beam submodels, and let N_Γ be the number of beam–joint interfaces. Let $n_{J,j}$ be the internal solid DOF count of joint model j. In the present implementation each interface contributes six coupling DOFs, so the global interface vector contains

$$n_\Gamma = 6N_\Gamma$$

unknowns. Because each beam has two coupled-interface ends, the formulation also gives

$$N_\Gamma = 2N_B.$$

This separation between internal joint DOFs and interface DOFs underlines the fact that refining the mesh inside one joint changes $n_{J,j}$, and therefore changes the cost of that local 3D solve, but it does not enlarge the global interface vector n_Γ.

Let $T_J(n_{J,j})$ and $M_J(n_{J,j})$ be the wall time and memory of one local joint solve with $n_{J,j}$ internal DOFs. Let T_B and M_B be the corresponding beam-model costs. One serial residual evaluation of the staggered hybrid problem may then be written as

$$T^{\text{hyb}}_{\text{eval,ser}} \approx \sum_{j=1}^{N_J} T_J(n_{J,j}) + \sum_{b=1}^{N_B} T_B + c_\Gamma N_\Gamma,$$

$$M^{\text{hyb}} \approx \sum_{j=1}^{N_J} M_J(n_{J,j}) + \sum_{b=1}^{N_B} M_B + c_u N_\Gamma,$$

where the constants c_Γ and c_u sum up interface assembly, residual construction, vector storage, and similar coupling-level costs.

If P processor cores are used and the local joint solves are distributed across them, then the residual-evaluation time should decrease toward the ideal bound

$$T^{\mathrm{hyb}}_{\mathrm{eval,par}}(P) \lesssim \frac{1}{P}\sum_{j=1}^{N_J} T_J(n_{J,j}) + \sum_{b=1}^{N_B} T_B + c_\Gamma N_\Gamma,$$

until load imbalance between cores and the non-parallel part of the algorithm become dominant. The total wall time of the hybrid solve can then be written as

$$T_{\mathrm{hyb}} \approx N_R T^{\mathrm{hyb}}_{\mathrm{eval}},$$

where N_R is the total number of residual evaluations required by the staggered nonlinear interface solver.

The direct-solver literature discussed above explains why smaller local solid problems are attractive, while the literature shows that local elasticity solves need not inherit the same severe scaling as a monolithic direct factorization if effective preconditioning is available [50–53]. However, the present hybrid implementation is not simply a collection of independent local solves; it is a staggered nonlinear coupled problem. Parallelisation reduces the cost of one residual evaluation, but the staggered coupling can partially offset this gain if it increases the number of residual evaluations needed for convergence. For that reason, the conclusion of these observation is limited to: the hybrid model is expected to benefit from both DOF reduction and parallel local solves, but the final gain must be verified experimentally for the coupled formulation used here.

4.7.4. Effect of Joint Resolution and Surrogate Replacement

To make the previous conclusion more explicit, consider the case in which all joint models share the same internal DOF count n_J and nonlinear solve iterations. The leading dependence of one hybrid residual evaluation on joint resolution is

$$T^{\mathrm{hyb}}_{\mathrm{eval,ser}} \sim N_J T_J(n_J), \qquad M^{\mathrm{hyb}} \sim N_J M_J(n_J),$$

if beam and interface costs are secondary. This makes the joint model a repeated cost driver: increasing the resolution of one joint is not a local decision at the whole-model level, because that same cost is paid at every joint.

The same reasoning gives the benefit of replacing a solid joint model by a reduced or statistical surrogate. If one surrogate call requires wall time T_{stat} and memory M_{stat}, then the approximate serial savings are

$$\Delta T_{\mathrm{ser}} \approx N_R N_J \left[T_J(n_J) - T_{\mathrm{stat}}\right], \qquad \Delta M \approx N_J \left[M_J(n_J) - M_{\mathrm{stat}}\right].$$

In the parallel case the time saving is bounded by the reduction in the slowest per-core workload rather than by the full serial sum, but the memory saving still accumulates over all replaced joint models.

4.7.5. Effect of Increasing the Number of Unit Cells and Interfaces

Increasing the specimen size changes both the number of repeated subproblems and the number of interfaces. For a BCC lattice, with N_x, N_y, and N_z being the numbers of unit cells in the three coordinate directions. The number of joint models in the metamaterial model is

$$N_J = (N_x + 1)(N_y + 1)(N_z + 1) + N_x N_y N_z,$$

because the corner joints are shared between neighboring cells, whereas each cell contributes one body-centered joint. The number of beam submodels is

$$N_B = 8 N_x N_y N_z,$$

and the number of interfaces becomes

$$N_\Gamma = 2N_B = 16 N_x N_y N_z.$$

Since each interface contributes six coupling DOFs, the interface vector size is

$$n_\Gamma = 6N_\Gamma = 96N_xN_yN_z.$$

For a cubic $N \times N \times N$ BCC specimen this simplifies to

$$N_J = (N+1)^3 + N^3, \qquad N_B = 8N^3, \qquad N_\Gamma = 16N^3, \qquad n_\Gamma = 96N^3.$$

The first few cubic specimen cases are

N	unit cells	N_J	N_B	N_Γ
1	1	9	8	16
2	8	35	64	128
3	27	91	216	432
4	64	189	512	1024

The important point is not only that N_Γ grows cubically with specimen size, but that the coupling model still grows only linearly with N_Γ. The stronger cost increase usually comes indirectly, because more unit cells also mean more beams and more joint subproblems. For fixed local beam and joint templates, specimen-size growth therefore acts primarily through repeated local solves rather than through a large global interface system.

4.7.6. Qualitative Experimental Verification

Three small numerical studies were carried out to check whether the qualitative scaling arguments above are reflected by the implemented hybrid solver. All models considered in these studies were BCC lattices with $r/l = 0.15$, circular nodes, and a prescribed compressive strain of 0.1 in the z-direction. The outward-facing interfaces were assigned Dirichlet conditions on z-displacement and all rotations, while the x- and y-displacements remained free; in addition, the interface attached to the $(0, 0, 0)$ joint was fully constrained. The purpose of these runs is not to fit an exact scaling law, but to verify which quantities appear to control wall time and memory in practice. In the discussion below, the reported interface-DOF count refers to the size of the free interface-position matrix after boundary conditions have been applied. Likewise, the reported numbers of joints and interfaces should be interpreted as the active submodels and active coupling quantities in the constrained problem that is actually solved, rather than as the purely topological counts of an unconstrained lattice generator.

The first study examined parallelism on the same one-unit-cell model, with 8 joint solves, 8 beam elements, and 16 interfaces, while increasing the CPU core count available from 1 to 5 to 10. The data is summarized in Table 4.2. The wall time decreased from 528 s in serial to 202 s on 5 cores and 171 s on 10 cores. This confirms the expected benefit of solving the local subproblems in parallel. At the same time, the speedup is clearly sub-ideal, which is also consistent with the analytical discussion. In this model there are only nine computational tasks available, namely eight joint solves and one grouped beam solve, so a tenth CPU core cannot provide further concurrency. In addition, the staggered nonlinear coupling and interface bookkeeping introduce serial work. The experiment therefore supports the statement that the hybrid decomposition is computationally attractive, but also shows that the attainable speedup is limited by task count, load balance, and the residual-based coupling procedure.

The second study examined joint refinement at fixed geometry and fixed parallelism. The model was run on 10 cores with the same 8 joints, 8 beams, 16 interfaces, and 112 free interface-position DOFs, while the joint mesh was refined from 124 mesh nodes (372 joint DOFs) to 274 mesh nodes (822 joint DOFs). The data are summarized in Table 4.3. The wall time increased from 171 s to 1041 s, while the peak memory increased from 0.18 GB to 0.28 GB. This is the clearest experimental confirmation of the analytical model developed earlier in the section. The coupling size remained unchanged, but the cost still increased dramatically because each repeated local three-dimensional joint solve became more expensive. In fact, the joint DOF count increased by a factor of

$$\frac{822}{372} \approx 2.21,$$

Table 4.2: Parallelism study for the one-unit-cell BCC model.

Quantity	1 core	5 cores	10 cores
Joint mesh nodes	124	124	124
Joint DOFs	372	372	372
Active interfaces	16	16	16
Free interface-position DOFs	112	112	112
Active joints	8	8	8
Beams	8	8	8
Wall time [s]	528	202	171
Peak memory [GB]	0.15	0.11	0.18

while the wall time increased by a factor of

$$\frac{1041}{171} \approx 6.09.$$

This increase is therefore much stronger than linear. If the local cost law introduced earlier in the section,

$$T_J(n_J) \sim n_J^p,$$

is used as a model for this comparison, the observed data correspond to an effective exponent

$$p_{\text{eff}} \approx \frac{\log(1041/171)}{\log(822/372)} \approx 2.3.$$

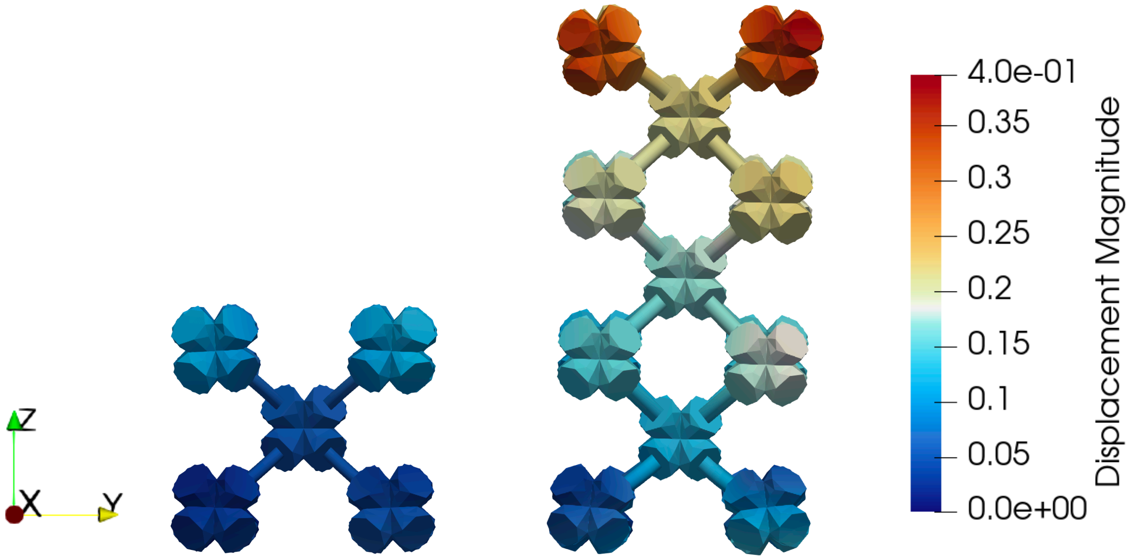
This value should not be interpreted as a universal fitted scaling exponent for the solver, since it is inferred from only two runs and from total wall time rather than isolated local solve time. However, it is fully consistent with the earlier thesis argument that local three-dimensional joint resolution is a dominant cost driver and that the repeated solid subproblems can exhibit strongly superlinear cost growth. In other words, the experiment supports the claim that joint resolution enters the whole-model cost primarily through repeated local solid solves, rather than through growth of the global interface vector.

Table 4.3: Joint-refinement study at fixed geometry on 10 cores.

Quantity	Coarse mesh	Fine mesh
Joint mesh nodes	124	274
Joint DOFs	372	822
Active interfaces	16	16
Free interface-position DOFs	112	112
Active joints	8	8
Beams	8	8
Wall time [s]	171	1041
Peak memory [GB]	0.18	0.28

Note also that the use of a simple Newton-Raphson nonlinear solver for the individual black-box solid element joint solves is impacted by larger DoF count models, because the linearized tangent solves become more expensive and because the nonlinear convergence can degrade with increasing resolution. An added number of necessary iterations can have an effect on wall time that is not captured by the simple cost model above. This makes solver choice in these black-box solves a impactful parameter for wall-time that was not optimised in this work.

The third study examined specimen-size growth while keeping the local joint discretization fixed. Here the model was increased from a single unit cell to a $1 \times 1 \times 3$ stack of three unit cells, still using 10 cores and the same per-joint mesh resolution. The compared model geometries are shown in Figure 4.3, and the numerical data are summarized in Table 4.4. Over this change, the number of joints increased from 9 to 19, the number of beams increased from 8 to 24, the number of interfaces increased from 16 to 48, and the number of free interface-position DOFs increased from 112 to 304. The wall time increased from 171 s to 233 s, while the peak memory increased from 0.18 GB to 0.44 GB. The time increase is therefore relatively mild compared with the growth in active interfaces, beams, and joints, whereas the memory increase is more pronounced. Qualitatively, this supports the view that interface growth by itself is not the dominant wall-time contribution for the present small models; rather, the main computational burden still appears to lie in the repeated local solves and in the nonlinear coupling procedure. At the same time, enlarging the specimen clearly increases the total amount of model state that must be stored, which is reflected in the memory growth.



(a) 1-unit cell model (b) 3-unit cell model

Figure 4.3: BCC models used in the specimen-size study: (a): a single unit cell and (b): a $1 \times 1 \times 3$ stack with identical local joint discretization both at final deformed state. The beam element struts are not shown to scale for visual clarity in separation of the subdomains.

Table 4.4: Specimen-size study at fixed local discretization on 10 cores.

Quantity	1 unit cell	$1 \times 1 \times 3$ unit cells
Joint mesh nodes	124	124
Joint DOFs	372	372
Active interfaces	16	48
Free interface-position DOFs	112	304
Active joints	9	19
Beams	8	24
Wall time [s]	171	233
Peak memory [GB]	0.18	0.44

This specimen-size study should also be interpreted carefully relative to the analytical counting formulas given earlier. Those formulas were written for nominal lattice counts in structured BCC assemblies, whereas the experimental model here is a constrained $1 \times 1 \times 3$ stack and the tabulated interface-position DOFs correspond only to free coupling quantities after boundary conditions. The experiment should therefore be read as validation of the qualitative trend that more unit cells produce more repeated subproblems and more coupling work, rather than as a point-by-point verification of the nominal unconstrained counting formulas.

Figure 4.4 summarizes the relative changes in wall time and peak memory across the three studies. Each panel corresponds to one experiment: parallelism, joint refinement, and specimen size. Within each panel, the values are normalized by the baseline case of that experiment, namely the 1-core run, the coarse joint mesh, and the single-unit-cell model, respectively. This normalization is useful because it allows the three experiments to be compared visually despite their different absolute scales.

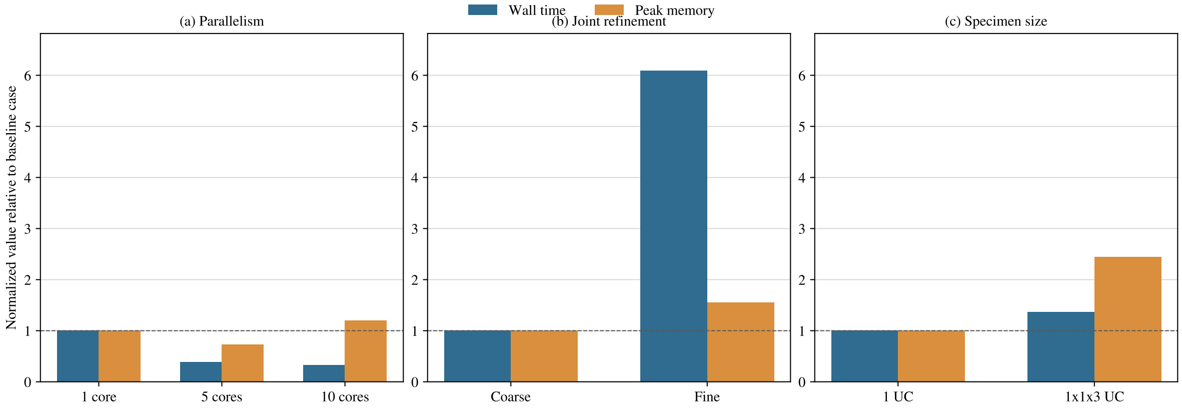


Figure 4.4: Normalized wall time and peak memory for the three qualitative scaling studies on the staggered hybrid element model. Panel (a) compares parallelism, panel (b) compares joint refinement, and panel (c) compares specimen size. In each panel, all values are normalized by the baseline case of that experiment.

Taken together, the three studies align well with the intended scaling picture of the hybrid model. First, local joint resolution is a strong cost driver. Second, parallel execution is beneficial, but its gain is bounded by the finite number of independent tasks and by the outer nonlinear coupling. Third, increasing the specimen size increases both total work and memory use, but in the present runs the interface number growth does not appear to be the dominant source of wall time. Since only total wall time and peak memory were recorded here, these observations are used as qualitative validation of the earlier analytical arguments rather than as a fitted predictive scaling law.

4.7.7. Main Implications

The analytical discussion and the qualitative experiments above lead to four scaling observations that are central for the remainder of this thesis:

1. In fully solid nonlinear lattice models, the total solid DOF count $N_{\text{dof}}^{\text{sol}}$ remains the natural analytical cost driver, because it affects every assembly step, every Newton iteration, and the size of every linear solve. The monolithic scaling arguments in this section should therefore still be read as the main motivation for replacing large solid domains by a hybrid description.
2. The strongest experimental result obtained here is that joint resolution is a dominant repeated cost driver in the partitioned hybrid model. Refining the solid joint mesh increased wall time strongly even though the free interface-position vector was unchanged, which supports the claim that local three-dimensional joint solves, rather than the interface vector size by itself, control a large part of the total computational cost.
3. Parallel execution of the local submodels clearly improves wall time, but the gain is limited by the finite number of available tasks and by the staggered nonlinear coupling. In the present one-unit-cell test case, there were only nine tasks in total, namely eight joint solves and one grouped beam solve, so adding a tenth core could not produce ideal additional speedup.
4. Increasing specimen size increases both memory usage and total computational work, but in the present small-scale experiments the growth of interface quantities did not appear to dominate wall time. The data therefore support a qualitative picture in which larger specimens mainly increase cost through more repeated local subproblems and more stored model state, while the precise balance between interface work and local solve work remains problem-dependent.

Note this section deliberately focused on discretization-level cost drivers, as the different interface solvers discussed earlier in this chapter mainly determine how many residual evaluations N_R are required; the study and points made here determine how expensive each of those evaluations is expected to be.

5

Mesh Convergence, Validation and Accuracy

Reduced-order modeling is introduced in this thesis to reduce the computational cost of nonlinear lattice simulations while retaining the influence of finite joint geometry. This reduction necessarily introduces modeling error, because the full three-dimensional continuum description of the lattice is replaced either partly, in the hybrid model, or entirely, in the beam-only models, by one-dimensional beam elements. The purpose of this chapter is therefore to establish the numerical credibility and practical accuracy of the modeling approaches before they are used to interpret joint-geometry effects.

The chapter distinguishes three levels of evidence. First, numerical convergence studies are used to verify that the selected solid-joint meshes and beam discretizations do not dominate the reported responses. Second, controlled validation cases are used to check the hybrid beam–3D continuum coupling against solid element references. Third, the complete hybrid formulation is assessed across the representative parameter space used in this thesis by comparing it with full solid element and pure beam element unit-cell models.

The main quantities used for comparison are the initial effective Young's modulus, E_0^*, the effective elastic energy absorption, and, where instability dominates the response, an apparent softening-onset strain. These metrics separate agreement in the near-linear regime from agreement in the nonlinear and post-buckling response. This distinction is important because a model may reproduce the initial stiffness while still predicting an inaccurate softening onset or post-buckling load level.

The structure of this chapter is as follows. Section 5.1 presents the interface validation, the solid-joint mesh convergence studies, and the beam discretization convergence studies. These checks establish that the later differences between modeling approaches are not dominated by obvious numerical discretization errors. Section 5.2 then evaluates the accuracy of the hybrid element model across architecture type, r/l, and joint geometry. The chapter concludes by identifying the regimes in which the hybrid model is most reliable, the cases where caution is required, and the extent to which it preserves joint-geometry trends at reduced computational cost.

5.1. Validation and Mesh Convergence of the Hybrid Element Model

This section establishes the numerical reliability of the hybrid formulation before the broader accuracy study. The interface validation checks the beam–solid coupling in a controlled single-strut problem. The convergence studies then determine the solid-joint and beam-strut discretizations needed to make the hybrid unit-cell response insensitive to numerical refinement. Separate convergence studies are also performed for the solid-element only and beam element only lattice unit cell models used in the accuracy as benchmarks Finally, a high-relative-density BCC unit-cell case is compared directly against a solid element reference under large deformation.

5.1.1. Single-Strut Interface Validation

The hybrid formulation couples continuum joint regions to beam elements through reduced interface degrees of freedom. This coupling introduces a modeling assumption: the deformation of the solid cross-section at the joint-side interface is not resolved freely, but is instead represented through the reduced kinematic description used to connect the joint to the beam strut. Since this constraint can introduce artificial stiffness, the interface treatment must be checked before the method is applied to full unit-cell simulations.

A single-strut interface validation problem is therefore used as a controlled comparison between a hybrid model and an equivalent solid element reference. The model consists of two octet-type joint regions connected by one strut. A geometrically equivalent solid element model is generated in CAD and meshed with tetrahedral elements. Both the hybrid and solid models are generated for $r/l = 0.10$ and $r/l = 0.03$ using circular joint geometry. The solid reference mesh is refined more strongly than the joint meshes used in the hybrid model, so that the comparison primarily evaluates the hybrid interface idealization rather than the solid mesh density.

One joint interface is fully clamped, while the opposite joint interface is prescribed a transverse displacement of $u_z = -0.2$ and constrained in the remaining degrees of freedom. The strut lies in the xy plane, so the imposed displacement creates a shear- and bending-dominated deformation of the single-strut system. The models and boundary conditions are shown in Figure 5.1. The hybrid variants are solved using the full-Jacobian Newton interface-equilibrium solver, which is feasible because of the small problem size.

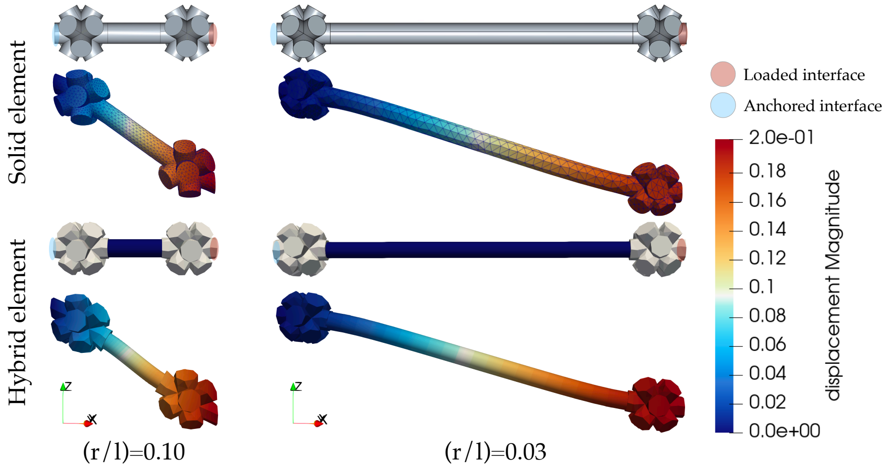


Figure 5.1: Single-strut interface validation against solid element reference models for $r/l = 0.10$ on the left and $r/l = 0.03$ on the right. From top to bottom: CAD geometry of the solid reference model, deformed solid element result with visible mesh, hybrid element setup, and deformed hybrid element result under imposed transverse displacement.

The comparison metric is the reaction force at the clamped side of the single-strut system. The results are summarized in Table 5.1. For the $r/l = 0.10$ model, the differences between the hybrid and solid reactions are 9.2%, 9.32%, and 19.07% for R_x, R_y, and R_z, respectively. For the $r/l = 0.03$ model, the corresponding differences are 10.48%, 10.48%, and 9.09%.

These differences are non-negligible, but the imposed displacement of $u_z = -0.2$ is intentionally large and drives the single-strut system into a strongly nonlinear deformation state. In the small-displacement regime, using an imposed displacement of $u_z = -0.01$, the largest disagreement between the hybrid and solid models is 0.31%. The interface validation therefore shows that the hybrid coupling behaves consistently in the low-strain regime, while larger deviations may appear under strong geometric nonlinearity. The full unit-cell accuracy studies later in this chapter are therefore required to quantify

Table 5.1: Reaction forces at the clamped side of the single-strut interface validation models.

Model	R_x	R_y	R_z
$r/l = 0.10$ solid element	-1.631	-1.631	1.694
$r/l = 0.10$ hybrid element	-1.781	-1.783	2.017
$r/l = 0.03$ solid element	-0.315	-0.315	0.143
$r/l = 0.03$ hybrid element	-0.348	-0.348	0.156

the modeling error at the deformation levels used in this thesis.

5.1.2. Mesh and Beam Discretization Convergence in the Hybrid Formulation

Within the hybrid formulation, the solid joint mesh must be converged because the joint regions are the only continuum parts of the reduced model and are also the regions where joint-geometry effects are introduced. The mesh convergence study is therefore performed before the full accuracy comparison, so that the later differences between modeling approaches are not dominated by joint mesh density.

Two demanding cases are selected for the joint mesh convergence study. The first is a stretching-dominated octet unit cell with circular joint geometry and $r/l = 0.03$. This case uses the smallest r/l considered in this thesis and is loaded into the geometrically nonlinear regime. The second is a bending-dominated pyramidal unit cell with circular joint geometry and $r/l = 0.10$. This case is included because the finite joint region occupies a larger fraction of the unit-cell geometry and is therefore expected to be particularly sensitive to the joint discretization.

The octet unit cell is subjected to an axial compressive strain of $\varepsilon_z = -0.1$. Dirichlet boundary conditions are applied to the reduced interface degrees of freedom. Interfaces on the z^+ face that are not coupled to struts are prescribed $u_z = -0.1$, while their remaining degrees of freedom remain free. Corresponding free interfaces on the z^- face are prescribed $u_z = 0$. One interface at the origin, oriented in the $(-1, -1, 0)$ direction, is fully constrained to remove rigid-body motion. These boundary conditions are illustrated in Figure 5.2. All stress–strain results in this section are reported using positive compressive magnitudes. The simulations are performed under axial compression, but the plotted strain and stress values are therefore presented as positive quantities to simplify comparison between the different modeling approaches.

The beam sections of the hybrid model are discretized with 10 elements along each strut, so that the influence of the solid joint mesh refinement can be isolated. The joint meshes used in the octet convergence study are summarized in Table 5.2. The Gmsh `clscale` parameter controls the characteristic element size in the generated unstructured mesh. The corresponding CAD model and representative meshes are shown in Figure 5.3.

Table 5.2: Models used for the solid joint mesh convergence study of the octet architecture with circular joint geometry and $r/l = 0.03$.

Gmsh `clscale` parameter	Number of mesh nodes	Number of elements
2	203	1089
1.5	219	1160
1.3	267	1452
1.2	280	1521
1	316	1741
0.9	382	2035

The simulations are run using 20 equal loading steps. The resulting effective engineering stress–strain curves are shown in Figure 5.4. The effective engineering stress is computed as

$$\sigma_z^* = \frac{F_z}{A_{UC}},$$

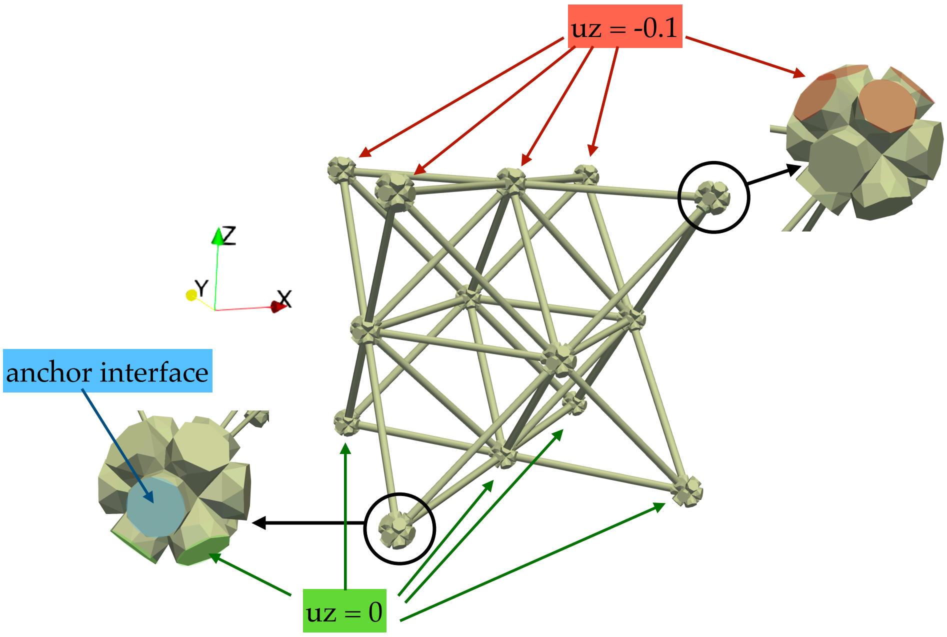


Figure 5.2: Boundary conditions applied to the reduced interfaces of the octet unit-cell models used for the hybrid joint mesh convergence study. The beam sections are shown schematically and are not drawn to scale.

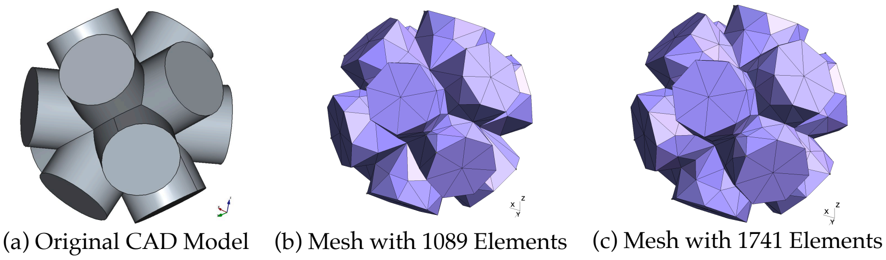

(a) Original CAD Model (b) Mesh with 1089 Elements (c) Mesh with 1741 Elements

Figure 5.3: Parametric CAD model and representative joint meshes used for the octet joint mesh convergence study.

where F_z is the reaction force in the z direction at the loading face and A_{UC} is the reference side area of the unit cell.

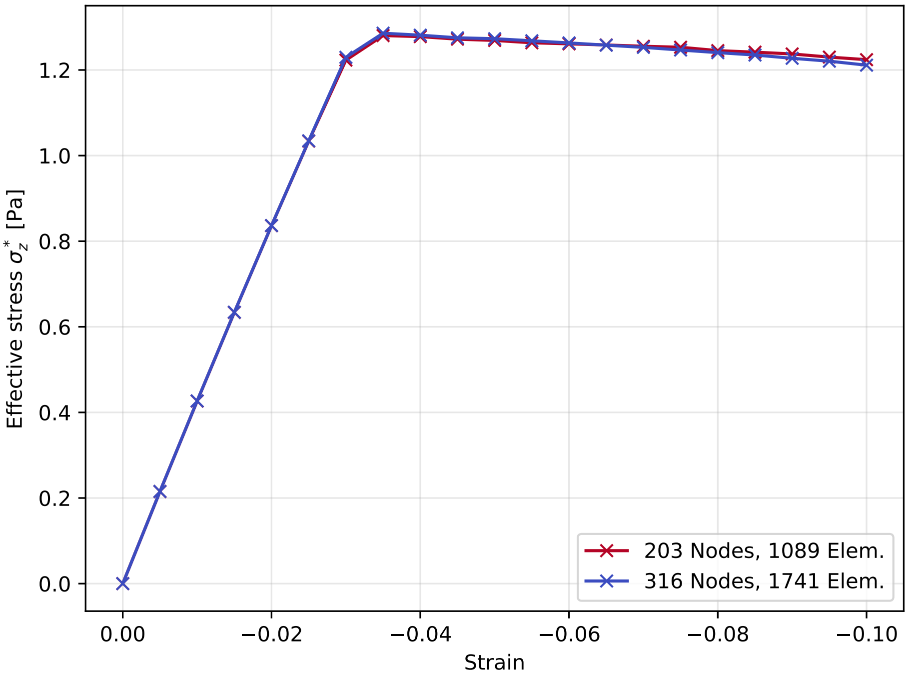


Figure 5.4: Joint mesh convergence study for the octet architecture with circular joint geometry and $r/l = 0.03$: effective engineering stress–strain curves for different joint mesh refinements. The results are reported using positive compressive magnitudes.

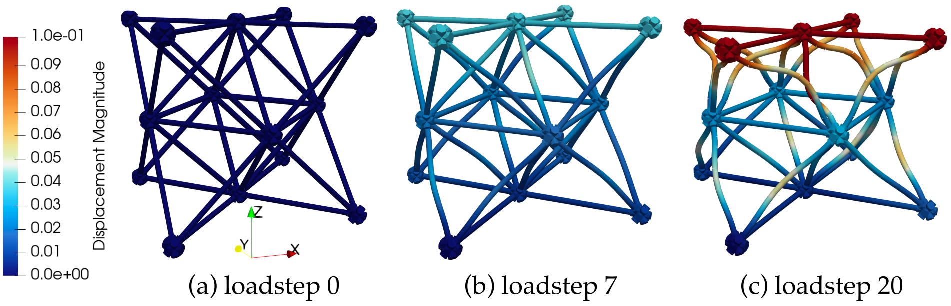
Table 5.3: Mesh convergence metrics for the octet unit cell with circular joint geometry and $r/l = 0.03$. The effective elastic energy absorption is computed by trapezoidal integration of the effective engineering stress–strain curve, while E_0^* is computed from the first nonzero loading step.

Number of mesh nodes	Effective elastic energy absorption	Initial effective Young's modulus, E_0^*
203	0.106628	43.0289
316	0.106551	43.0697

The effective engineering stress–strain curves in Figure 5.4 are nearly mesh-independent over the full loading path. This is confirmed by the scalar metrics in Table 5.3. The difference in E_0^* between the coarsest and finer mesh is 0.07%, which indicates that the near-linear response is insensitive to the joint mesh refinement. Only the first nonzero load step is used for this modulus calculation in order to avoid including the onset of nonlinear response.

The nonlinear part of the response is assessed using the effective elastic energy absorption, computed as the area under the effective engineering stress–strain curve. This quantity is used because it measures the path-integrated mechanical work absorbed by the unit cell under the prescribed deformation, including the effect of geometrically nonlinear deformation. Since the material model used in this chapter is elastic, the quantity should be interpreted as elastic work along the prescribed deformation path rather than as plastic energy dissipation. Table 5.3 shows a difference of only 0.095% between the two meshes, confirming that the 203-node joint mesh is sufficient for the octet case. Deformed results obtained using this mesh are shown in Figure 5.5.

A second mesh convergence study is performed on a pyramidal unit cell with circular joint geometry and $r/l = 0.10$. This case provides a complementary bending-dominated validation problem in which



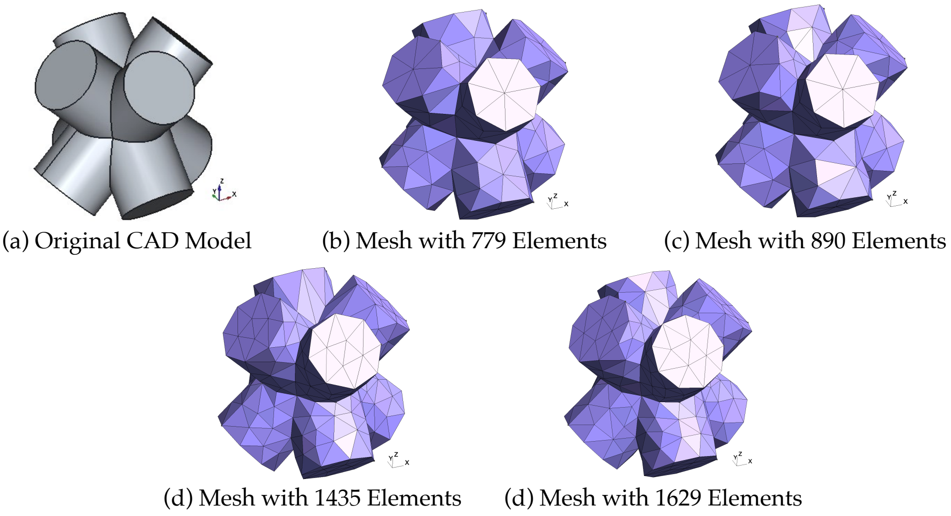
(a) loadstep 0 (b) loadstep 7 (c) loadstep 20

Figure 5.5: Deformed simulation results for the octet unit cell with $r/l = 0.03$ using the 203-node joint mesh. The configurations are shown at load step 0, load step 7 immediately after the stress–strain curve kink, and load step 20 at the final strain $\varepsilon_z = -0.1$.

the joint region constitutes a larger fraction of the unit-cell volume. Since the pyramidal architecture exhibits a less abrupt nonlinear response than the octet case, the model is loaded to an axial compressive strain of $\varepsilon_z = -0.4$. The joint meshes used in this study are summarized in Table 5.4, and representative meshes are shown in Figure 5.6.

Table 5.4: Models used for the solid joint mesh convergence study of the pyramidal architecture with circular joint geometry and $r/l = 0.10$.

Gmsh `clscale` parameter	Number of mesh nodes	Number of elements
1.5	155	779
1.3	176	890
1	276	1435
0.9	317	1629



(a) Original CAD Model (b) Mesh with 779 Elements (c) Mesh with 890 Elements

(d) Mesh with 1435 Elements (d) Mesh with 1629 Elements

Figure 5.6: Parametric CAD model and representative joint meshes used for the pyramidal joint mesh convergence study.

The effective engineering stress–strain curves for the pyramidal mesh convergence study are shown in Figure 5.7. The curves remain close over the full loading path, indicating that the joint mesh refinement has a limited influence on the global response for the range of meshes tested.

Deformed results of the selected coarse-mesh simulation are shown in Figure 5.8 at three load steps.

The largest difference in effective elastic energy absorption between the finest mesh and the three coarser meshes is 1.56%. For E_0^*, the largest difference is 2.73%, while the difference between the two finest meshes is 1.46%. These differences indicate near convergence, and the mesh consisting of 276

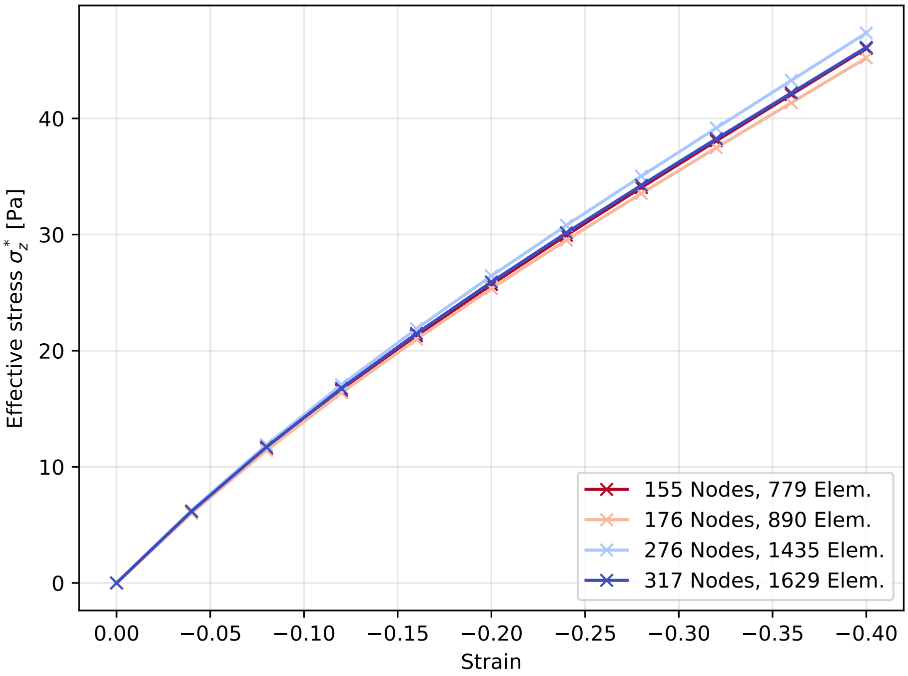
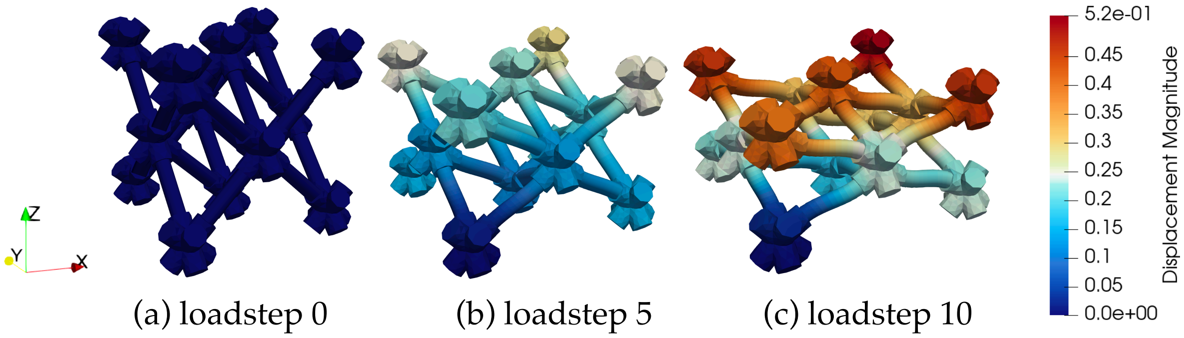


Figure 5.7: Joint mesh convergence study for the pyramidal architecture with circular joint geometry and $r/l = 0.10$: effective engineering stress–strain curves for different joint mesh refinements. The results are reported using positive compressive magnitudes.



(a) loadstep 0 (b) loadstep 5 (c) loadstep 10

Figure 5.8: Deformed simulation results for the pyramidal unit cell with $r/l = 0.10$ using the 276-node joint mesh. The configurations are shown at load step 0, load step 5 at $\varepsilon_z = -0.2$, and load step 10 at the final strain $\varepsilon_z = -0.4$.

nodes and 1435 elements is therefore considered sufficient for the following validation and accuracy studies. This mesh refinement point is extended to the square-joint meshes, where a similar or higher number of degrees of freedom is used for all cases.

Finally, the discretization of the beam elements forming the connecting struts is considered. In the previous convergence studies, all beam struts were discretized using 10 elements along the member length. To verify that this discretization does not influence the results significantly, the converged octet and pyramidal models are rerun using different numbers of beam elements per strut. The results are summarized in Table 5.5.

Table 5.5: Beam discretization convergence metrics for the octet and pyramidal hybrid models. The effective elastic energy absorption is computed by trapezoidal integration of the effective engineering stress–strain curve, while the Initial effective Young's modulus, E_0^* is computed from the first nonzero loading step.

Model	Number of beam elements per strut	Effective elastic energy absorption	E_0^*
Octet	40	0.104885	42.3507
Octet	20	0.105057	42.5741
Octet	10	0.106628	43.0289
Octet	5	0.105398	43.9652
Pyramidal	10	10.2206	156.5304
Pyramidal	5	10.2364	156.6764

For the pyramidal case, 5 beam elements per strut are sufficient: the change from 10 to 5 elements is 0.15% in effective elastic energy absorption and 0.09% in E_0^*. The octet case is more sensitive to beam discretization. A reasonable level of mesh independence is obtained at 20 beam elements per strut, with a 0.16% change in effective elastic energy absorption and a 0.52% change in E_0^* relative to the 40-element result.

The following validation and accuracy studies therefore use the converged solid joint meshes identified above and a beam discretization of 20 elements per strut for all hybrid formulation models.

5.1.3. Beam Discretization Convergence in Pure Beam Element Models

The accuracy study in Section 5.2 uses pure beam element models as lower-fidelity references. These models are not hybrid models: the finite joint regions are collapsed into zero-volume beam-network nodes, and the complete unit cell is represented by beam elements. A separate convergence check is therefore required to ensure that the beam-only results are not controlled by the number of beam elements used along each strut.

Two representative Timoshenko beam cases are selected. The first is the pyramidal unit cell with $r/l = 0.10$, which is bending-dominated and later shows the largest difference between beam-only and joint-resolved models. The second is the octet unit cell with $r/l = 0.03$, which is stretching-dominated and exhibits an instability-dominated nonlinear response. Both cases are analyzed using the same boundary conditions as the corresponding pure beam models in the accuracy study.

Table 5.6 shows that the initial effective Young's modulus of the pure beam models is essentially insensitive to the number of beam elements per strut for both representative cases. For the pyramidal $r/l = 0.10$ case, the effective elastic energy absorption changes by only approximately 0.03% between 2 and 20 elements per strut. The pure beam response of this bending-dominated unit cell is therefore already converged for the global scalar metrics at the coarsest tested discretization.

The octet $r/l = 0.03$ case shows a different behavior. The initial effective Young's modulus is again mesh-independent, but the apparent buckling onset strain changes from $\varepsilon_{\text{onset}} = 0.078$ with 2 elements per strut to $\varepsilon_{\text{onset}} = 0.050$ with 5 or more elements per strut. The apparent onset strain used $\varepsilon_{\text{onset}}$ is the strain at which the effective engineering stress falls 10% below the initial linear extrapolation $\sigma^* = E_0^*\epsilon$. This indicates that 2 elements per strut are sufficient for the near-linear stiffness, but not for the instability-related metric. Since the onset strain and initial stiffness are unchanged for 5, 10, and

Table 5.6: Beam discretization convergence for representative pure Timoshenko beam element unit-cell models. The pyramidal $r/l = 0.10$ case is evaluated using E_0^* and effective elastic energy absorption. The octet $r/l = 0.03$ case is evaluated using E_0^* and the apparent softening-onset strain $\varepsilon_{\text{onset}}$

Case	Elements per strut	E_0^*	Effective elastic energy absorption	$\varepsilon_{\text{onset}}$
Pyramidal, $r/l = 0.10$	2	71.131183	1.244423	–
Pyramidal, $r/l = 0.10$	5	71.131178	1.244108	–
Pyramidal, $r/l = 0.10$	10	71.131178	1.244071	–
Pyramidal, $r/l = 0.10$	20	71.131178	1.244062	–
Octet, $r/l = 0.03$	2	40.216434	–	0.078
Octet, $r/l = 0.03$	5	40.216387	–	0.050
Octet, $r/l = 0.03$	10	40.216384	–	0.050
Octet, $r/l = 0.03$	20	40.216384	–	0.050

20 elements per strut, 5 elements per strut are used for all pure beam element unit-cell models in the accuracy study.

The convergence check is performed for the Timoshenko formulation, however, the same element count is extended to the Euler–Bernoulli beam references in the accuracy study.

5.1.4. Mesh Convergence in Pure Solid Element Models

The pure solid element models are used as reference solutions in the accuracy study. Their mesh convergence must therefore be checked separately, so that the reported differences between the solid, hybrid, and beam-only models are not dominated by unresolved solid mesh refinement. Since fully solid unit-cell simulations are substantially more expensive than the reduced models, the convergence study is limited to two representative solid element cases and focuses on the initial effective Young's modulus.

The first case is the pyramidal unit cell with circular joints and $r/l = 0.03$. This case is selected because it represents a slender bending-dominated architecture, for which the solid mesh must resolve both thin struts and finite joint regions. The second case is the octet unit cell with square joints and $r/l = 0.10$. This case is selected because it represents a thicker stretching-dominated architecture with a modified joint geometry. Both convergence studies are performed using the same boundary-condition strategy as the corresponding solid element models in the accuracy study.

Table 5.7: Mesh convergence of representative pure solid element unit-cell models. The Gmsh `clscale` parameter controls the characteristic element size used for mesh generation. The convergence ratio is computed relative to the finest mesh available for each case. The reported E^* values are obtained from a linear initial-stiffness calculation and may therefore differ slightly from the initial stiffness values reported in the nonlinear accuracy study.

Case	Gmsh `clscale`	Number of mesh nodes	Number of DOFs	E^*
Pyramidal, $r/l = 0.03$, circular	0.04	78035	234105	0.851
Pyramidal, $r/l = 0.03$, circular	0.05	46479	139437	0.874
Pyramidal, $r/l = 0.03$, circular	0.07	21755	65265	0.953095
Octet, $r/l = 0.10$, square	0.15	28053	84159	567.70555
Octet, $r/l = 0.10$, square	0.20	18026	54078	572.32800

Table 5.7 shows that the pyramidal $r/l = 0.03$ circular-joint model is more sensitive to solid mesh refinement than the octet $r/l = 0.10$ square-joint model. For the pyramidal case, the intermediate mesh with `clscale`= 0.05 differs from the finest mesh by approximately 2.7% in initial stiffness, whereas the coarsest mesh differs by approximately 12.0%. The intermediate mesh is therefore selected as the converged reference for this case. For the octet $r/l = 0.10$ square-joint case, the coarser `clscale`= 0.20 mesh differs from the finer `clscale`= 0.15 mesh by approximately 0.8%, and is therefore considered sufficient for the global initial-stiffness comparison.

The convergence tolerances used here are broader than those used for the reduced beam discretization because the fully solid models are considerably more expensive. The purpose of this check is therefore not to establish exact mesh-independent continuum solutions, but to verify that the selected solid references are sufficiently resolved for the global comparison metrics used in this thesis. More refined solid-element convergence studies would be required to extract more precise absolute accuracy errors for the hybrid formulation given larger computational resources.

The remaining solid reference models in the accuracy study are meshed at comparable or higher degrees of freedom than the selected converged meshes for cases with similar r/l and joint geometry complexity. This limits the number of full solid convergence studies required while still ensuring that the solid references used in the model comparison are not based on obviously under-resolved meshes. Consequently, the accuracy differences reported later in this chapter should be interpreted primarily as modelling and boundary-condition differences between the solid, hybrid, and beam-only formulations, rather than as artifacts of gross solid mesh under-resolution.

5.1.5. Validation of the Hybrid Formulation against Solid Element Models

Before the broader parametric accuracy study is performed, a high-relative-density unit-cell validation case is used to compare the hybrid model directly against a solid element reference under large deformation. This validation case is a body-centered cubic lattice unit cell with $r/l = 0.15$. At this value, the joint regions occupy up to 75% of each strut length, making it a demanding case for the hybrid formulation.

The boundary conditions are similar to those described in Figure 5.2, but rotations are additionally constrained on the prescribed boundary interfaces. The validation metrics are the average reaction force in the z direction, R_z, and the maximum displacement differences in the x and y directions, which provide a measure of the lateral deformation response. These quantities are evaluated at four levels of applied compressive strain: $u_z = -0.01$, $u_z = -0.1$, $u_z = -0.16$, and $u_z = -0.3$. The simulation results are presented in Table 5.8.

Table 5.8: Validation metrics R_z, u_x, and u_y obtained from the solid element model (sol.) and hybrid element model (hyb.) of a BCC unit cell under different prescribed axial compressive strains.

Applied u_z strain	R_z hyb.	R_z sol.	u_x hyb.	u_x sol.	u_y hyb.	u_y sol.
-0.01	0.7262	0.7626	7.0 e-3	6.7 e-3	7.0 e-3	6.7 e-3
-0.1	6.4106	6.7412	7.1 e-2	6.7 e-2	7.0 e-2	6.8 e-2
-0.16	9.4479	9.8701	1.12 e-1	1.07 e-1	1.10 e-1	1.10 e-1
-0.3	14.6817	14.5843	2.01 e-1	2.04 e-1	1.99 e-1	2.14 e-1

The BCC validation case shows close agreement between the hybrid and solid element models over the deformation range considered. The reaction force agreement remains good even at the largest applied strain, where the hybrid model gives $R_z = 14.6817$ and the solid element model gives $R_z = 14.5843$. The lateral displacement measures are also comparable, although they should be interpreted with the same caution as the later unit-cell comparisons because the solid and hybrid formulations do not impose rotational constraints in exactly the same way.

This validation case also illustrates the computational motivation for the hybrid approach. The solid element model requires a peak memory usage of 30.52 GB, while the corresponding hybrid model requires 1.43 GB. The broader performance implications were discussed in section 4.7. In this chapter, this result motivates the more systematic accuracy study against solid element references.

5.2. Accuracy of the Hybrid Element Model across the Parametric Space

The preceding sections establish that the hybrid formulation is mesh-converged and that the beam–solid interface behaves consistently in controlled validation cases. The remaining question is whether the method remains accurate when applied to the nonlinear unit-cell simulations of interest in this thesis. This section therefore compares hybrid models against mesh-converged solid element references across a limited but representative parameter space: two architectures, two values of r/l, and two joint geometries.

The comparison is not intended as a full parametric sweep. Instead, it targets the main regimes encountered in the following chapters: slender versus thick struts, stretching- versus bending-dominated deformation, and circular versus square joint transitions. Beam-only Euler–Bernoulli and Timoshenko models are included as lower-fidelity references, so that the improvement obtained by retaining continuum joint regions in the hybrid model can be assessed directly. In total, 24 models are compared: 12 octet models and 12 pyramidal models.

This section first presents the modeling and boundary-condition strategies used for the comparison, then shows the resulting stress–strain curves and scalar accuracy metrics for all cases. Finally, the results are interpreted in terms of the relative influence of modeling idealizations, boundary-condition differences, and discretization errors on the observed discrepancies between the solid, hybrid, and beam-only models.

5.2.1. Models, Boundary Conditions and Loading Conditions

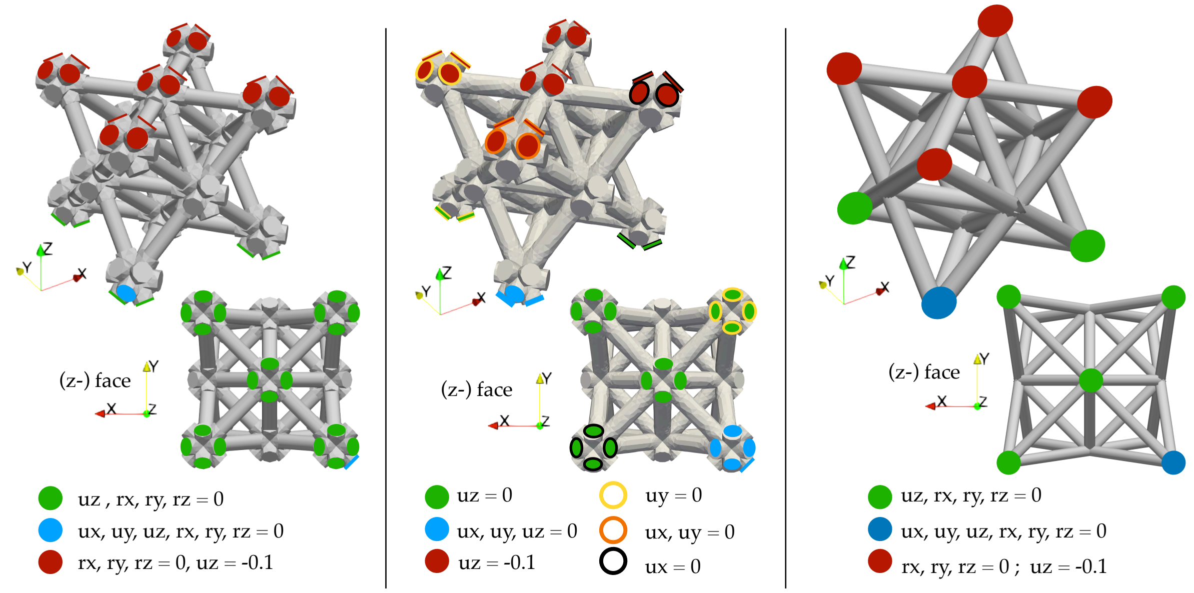
The model-accuracy study compares solid element, hybrid element, and beam-only representations of the same lattice unit-cell configurations. Two architectures are considered: an octet unit cell, representing a stretching-dominated architecture, and a pyramidal unit cell, representing a bending-dominated architecture. Each architecture is evaluated at $r/l = 0.03$ and $r/l = 0.10$, with both circular and square joint geometries.

All simulations are performed under axial compression. Similarly to the previous section, the stress–strain curves and scalar accuracy metrics in this section are reported using positive compressive stress and strain magnitudes. The octet models are loaded to a compressive strain magnitude of 0.1, while the pyramidal models are loaded to a compressive strain magnitude of 0.2. The effective engineering stress is computed from the total reaction force in the z direction divided by the undeformed unit-cell footprint area. Since the footprint is 1×1, this is numerically equal to the total reaction force. The resulting values are therefore intended for comparison between modeling approaches, rather than as fully periodic homogenized material properties.

The hybrid beam–3D continuum model defines the target interface-level loading case, as shown in Figure 5.9(a). The hybrid unit cells extend beyond the nominal $1 \times 1 \times 1$ domain because the solid joint regions protrude past the unit-cell boundaries. Compression is applied through the outer joint interfaces on the z^+ face, while the corresponding interfaces on the z^- face are constrained with $u_z = 0$. All prescribed boundary interfaces are also constrained in rotation. One lower boundary interface is fully fixed and acts as an anchor point to remove rigid-body motion.

The beam-only models use the same reduced kinematic prescription as the hybrid model, as shown in Figure 5.9(c), but apply it to an idealized lattice skeleton. Since the beam-only models do not include finite joint volumes extending past the unit-cell boundaries, the displacement and rotational constraints are applied directly at the boundary joint nodes of the $1 \times 1 \times 1$ beam network. The beam-only and hybrid boundary conditions are therefore equivalent only in a reduced kinematic sense: the same nominal joint motions are imposed, but the local boundary stiffness and effective load-transfer length are not identical.

The solid element models retain the same external geometry as the hybrid models, including the joint regions extending beyond the nominal unit-cell boundaries. The same boundary joint faces are selected for loading, as shown in Figure 5.9(b). However, the solid formulation has only translational degrees of freedom at each mesh node, (u_x, u_y, u_z), and no rotational degrees of freedom. As a result, the rotational constraints used in the hybrid and beam-only models cannot be imposed explicitly. Instead, the compression test is approximated by prescribing uniform u_z values on the selected boundary face patches, together with one fully fixed anchor face. Additional in-plane constraints are introduced on selected boundary joint faces on the x^- and y^- sides to reduce residual global twisting of the specimen.



(a) hybrid element model (b) solid element model (c) beam element model

Figure 5.9: Boundary conditions used in the accuracy study for (a) the hybrid element model, (b) the solid element model, and (c) the beam-only model. The colored markers indicate constrained boundary joints or boundary joint faces. The imposed compressive displacement shown schematically as $u_z = -0.1$ corresponds to the octet cases; for the pyramidal cases, the same loading pattern is used with $u_z = -0.2$.

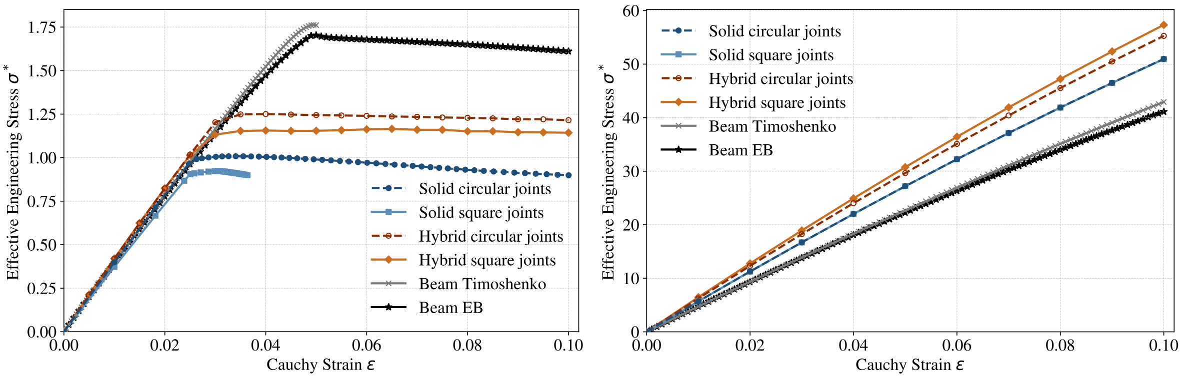
The three model classes are therefore compared under boundary conditions that are deliberately made as similar as possible, but are not mathematically identical. The hybrid model defines the target interface-level loading case, because it naturally prescribes six-component interface motions containing both translations and rotations. The beam-only model reproduces this loading case on an idealized zero-volume lattice skeleton. The solid element model, however, has only translational nodal degrees of freedom and must approximate the same loading case through prescribed displacements on finite boundary face patches. Rotational constraints and boundary-joint motions are therefore not represented in exactly the same way in the solid, hybrid, and beam-only models.

Consequently, the discrepancies reported in the accuracy study should not be interpreted as pure discretization errors of the hybrid formulation alone. They combine several effects: beam/solid modeling idealization, rigid-face interface assumptions, finite joint representation, and non-identical boundary-condition implementation. This distinction is especially important because the comparison is performed under geometrically nonlinear compression, where small differences in boundary rotation, local constraint stiffness, and effective load-transfer length can affect not only the absolute force level, but also the apparent softening onset and post-buckling path.

5.2.2. Results of the Model Accuracy Study

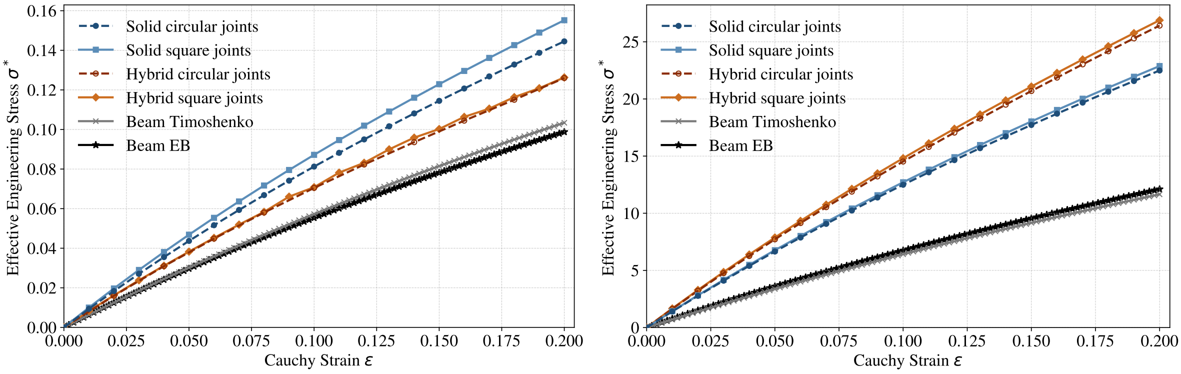
Figure 5.10 compares the effective engineering stress–strain response of the solid element, hybrid element, and beam-only models for the four representative validation cases. The agreement between the modeling approaches depends strongly on both architecture type and r/l. The thin cases, with $r/l = 0.03$, generally show better agreement in the initial regime because the struts have higher aspect ratio and the assumptions of beam theory are more appropriate. In contrast, the thick cases, with $r/l = 0.10$, show larger differences because the finite joint regions occupy a larger fraction of the unit-cell geometry. Some curves do not reach the target end strain because of nonlinear solver non-convergence. The non-smooth portions of some curves are shown in the figure for qualitative comparison, but are not completely converged (GMRES iteration timeout close to convergence) and not counted towards the accuracy metrics.

The octet case with $r/l = 0.03$, shown in Figure 5.10(a), is the clearest example of why agreement



a) Octet r/l = 0.03

b) Octet r/l = 0.10



c) Pyramidal r/l = 0.03

d) Pyramidal r/l = 0.10

Figure 5.10: Effective engineering stress–strain curves for the four unit-cell validation cases used in the accuracy study, plotted using positive compressive stress and strain magnitudes: (a) octet, r/l = 0.03; (b) octet, r/l = 0.10; (c) pyramidal, r/l = 0.03; and (d) pyramidal, r/l = 0.10. Solid element, hybrid element, Timoshenko beam, and Euler–Bernoulli beam models are compared for each case.

in the initial effective Young's modulus alone is not sufficient to validate a model for nonlinear lattice response. All modeling approaches begin with similar initial slopes, indicating reasonable agreement in the near-linear regime. However, the curves diverge after the onset of geometric softening. The beam-only models delay the apparent softening onset and carry higher stresses into the post-buckling regime, whereas the hybrid models remain closer to the corresponding solid element references. This indicates that the hybrid description is more capable of capturing the nonlinear influence of finite joint geometry in this slender, stretching-dominated architecture.

The deformed configurations in Figure 5.11 indicate that, after the onset of softening, the initially stretching-dominated octet response involves large rotations and bending of individual struts. This change in local deformation mode helps explain why agreement in the initial axial-stiffness regime does not carry directly into the post-buckling response. This interpretation is consistent with the expected mechanics of local strut buckling, where axial load transfer gives way to large rotations and bending of individual members after instability.

The octet case with $r/l = 0.10$, shown in Figure 5.10(b), exhibits a different trend. The hybrid models are stiffer than the solid element references over the full loading path, while the beam-only models remain softer. This suggests that the hybrid interface and solid-joint idealization recover much of the stiffness lost by the beam-only representation, but can also overconstrain or over-stiffen the response in thick-strut cases. The discrepancy should therefore not be interpreted only as a material or strut-discretization error, but also as a consequence of the finite joint idealization and the non-identical boundary-condition implementation between the solid and hybrid models.

The pyramidal cases show the strongest dependence on bending-dominated deformation. For $r/l = 0.03$, shown in Figure 5.10(c), the ordering of the curves is generally solid element, hybrid element, and then beam-only. The hybrid model therefore remains closer to the solid reference than the beam-only idealization, but still underestimates the effective response. This is consistent with the fact that the finite joint geometry still contributes to the stiffness of the bending-dominated unit cell, even though the struts are relatively slender.

The pyramidal case with $r/l = 0.10$, shown in Figure 5.10(d), is the clearest bending-dominated and joint-sensitive case. Here, the beam-only models recover only about half of the solid and hybrid initial response. This large discrepancy is consistent with the low aspect ratio of the struts and the increased relative size of the joint regions. A beam-only model idealises the lattice as line elements connected at zero-volume nodes, which removes the local stiffening and load-transfer effects of the finite joint geometry. The hybrid model reduces this error by retaining a continuum description of the joints while replacing only the connecting struts by beam elements.

The generally softer response of the beam-only models can also be interpreted in terms of the effective bending span. In the solid and hybrid models, finite joint regions reduce the length of the strut segment that deforms as a slender member. This is especially important in bending-dominated mechanisms, because the transverse stiffness of a beam segment scales strongly with its free length. For example, for a cantilever beam with end force F,

$$\delta = \frac{FL^3}{3EI}, \qquad k_b = \frac{F}{\delta} = \frac{3EI}{L^3},$$

where L is the effective bending span, E is Young's modulus, and I is the second moment of area. The cubic dependence on L means that even a moderate reduction in effective free span can significantly increase bending stiffness. The pure beam model instead represents the lattice as a skeleton of members connected at zero-volume nodes, so it does not retain the effective shortening and local rotational stiffening introduced by the finite joint geometry. This explains why the beam-only models are particularly soft in the pyramidal $r/l = 0.10$ case, where joint regions occupy a large fraction of the unit-cell geometry.

The similarity between the Euler–Bernoulli and Timoshenko beam results in Figure 5.10 and the data in Table 5.9 supports this interpretation. In most cases, both beam formulations give comparable values of E_0^* and effective elastic energy absorption. Even in the pyramidal $r/l = 0.10$ case, where their difference is largest, both beam-only models remain far below the solid and hybrid responses. This indicates that the main limitation of the pure beam models is not the absence of shear deformation, but the representation of finite joint regions as zero-volume nodes.

In addition to the effective stress–strain comparisons, representative deformed configurations are shown to clarify the modeling idealizations used in the accuracy study. Figure 5.11 illustrates how the same class of lattice response is represented at three levels of fidelity. In the solid element model, both joints and struts are resolved with continuum elements; in the hybrid model, the joint regions remain solid while the connecting struts are represented by beam elements; and in the beam-only model, the unit cell is reduced to a skeletal beam network. The selected octet and pyramidal cases provide visual references for the two architectures and span the range from a slender circular-joint configuration to a thicker square-joint configuration.

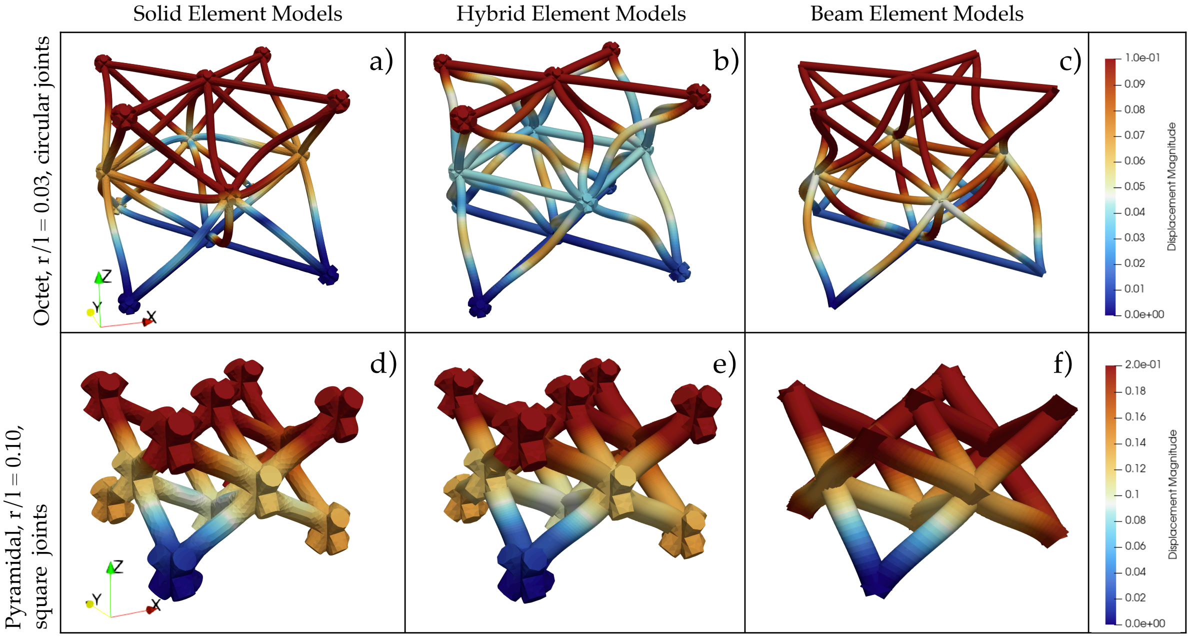


Figure 5.11: Representative deformed configurations of the finite element models used in the accuracy study. The columns distinguish the three modeling approaches: solid element models, hybrid solid-joint/beam-strut models, and beam-only models. The upper row shows the octet unit cell with circular joints and $r/l = 0.03$, while the lower row shows the pyramidal unit cell with square joints and $r/l = 0.10$. Panels (a) and (d) show the solid element models, panels (b) and (e) show the corresponding hybrid models, and panels (c) and (f) show the beam-only models. The octet beam-only model in panel (c) is based on Euler–Bernoulli beam theory, whereas the pyramidal beam-only model in panel (f) is based on Timoshenko beam theory. The color maps indicate displacement magnitude; separate color scales are used for the two rows.

The stress–strain curves in Figure 5.10 provide a qualitative comparison of the model responses. To quantify these differences more clearly, two scalar measures are extracted. The first is the initial effective Young's modulus, E_0^*, computed from the slope between the origin and the first nonzero plotted point. This quantity measures agreement in the near-linear effective response of the unit cell. The second is the effective elastic energy absorption, computed as the area under the effective engineering stress–strain curve using trapezoidal integration. Since the material model is elastic throughout this chapter, this quantity represents the path-integrated mechanical work absorbed by the unit cell under the prescribed deformation path, including geometrically nonlinear effects.

Table 5.9 confirms the main trends visible in Figure 5.10. For the octet unit cell with $r/l = 0.03$, all models predict similar values of E_0^*, which confirms that the early response of this slender architecture is captured reasonably well even by the beam-only models. However, this agreement does not extend to the nonlinear response, as shown by the delayed softening of the beam-only curves in Figure 5.10(a).

For the thick octet and thick pyramidal cases, the hybrid models overpredict E_0^* relative to the solid

Table 5.9: Initial effective Young's modulus, E_0^*, obtained from the effective engineering stress–strain curves in Figure 5.10. The value is computed as the slope from the origin to the first nonzero plotted point of each curve. Positive compressive stress and strain magnitudes are used.

Case	Solid circ.	Solid sq.	Hybrid circ.	Hybrid sq.	Beam Tim.	Beam EB
Octet, $r/l = 0.03$	39.9149	37.2765	42.3507	42.6583	40.2377	40.1977
Octet, $r/l = 0.10$	569.1000	569.2680	623.2190	646.2460	476.0780	478.7960
Pyramidal, $r/l = 0.03$	0.9342	1.0008	0.8001	0.8306	0.6388	0.6434
Pyramidal, $r/l = 0.10$	140.6870	143.1060	163.3210	166.6770	71.1312	77.3911

element references, while the beam-only models underpredict it. The largest beam-only discrepancy is observed for the pyramidal unit cell with $r/l = 0.10$, where the beam models recover only about half of the solid and hybrid initial response. In contrast, for the pyramidal unit cell with $r/l = 0.03$, the hybrid models remain closer to the solid references than the beam-only models, but still underpredict E_0^*. These trends indicate that the hybrid model does not behave as a uniformly softer or stiffer approximation; instead, its error depends on architecture type, relative joint size, and deformation mode.

Table 5.10: Effective elastic energy absorption obtained by trapezoidal integration of the effective engineering stress–strain curves in Figure 5.10. Positive compressive stress and strain magnitudes are used. The octet $r/l = 0.03$ case is omitted because the curves do not remain equally comparable over the full post-buckling strain range.

Case	Solid circ.	Solid sq.	Hybrid circ.	Hybrid sq.	Beam Tim.	Beam EB
Octet, $r/l = 0.10$	2.66125	2.66056	2.89652	3.00426	2.22613	2.17069
Pyramidal, $r/l = 0.03$	0.01566	0.01681	0.01357	0.01375	0.01108	0.01069
Pyramidal, $r/l = 0.10$	2.41585	2.45812	2.81831	2.87229	1.24406	1.30361

The same ordering is reflected in the effective elastic energy absorption values reported in Table 5.10. For the thick octet and thick pyramidal cases, the hybrid models absorb more energy than the corresponding solid element references, while the beam-only models remain below both. For the thin pyramidal case, the ordering becomes solid element, hybrid element, and then beam-only, indicating that the hybrid model captures part of the stiffness and energy absorption missing from the beam-only idealization, but does not fully recover the solid element response.

The octet unit cell with $r/l = 0.03$ is treated separately because several models enter a clear instability regime and some simulations do not converge sufficiently far beyond that point for a full-curve energy comparison to be meaningful. Instead of comparing effective elastic energy absorption, Table 5.11 quantifies the onset of geometric softening through the strain $\varepsilon_{\text{onset}}$ at which the effective engineering stress first falls 10% below the initial linear extrapolation $\sigma^* = E_0^* \varepsilon$. This quantity is used as an apparent softening-onset measure and should not be interpreted as an exact bifurcation strain.

Table 5.11: Softening-onset comparison for the octet unit cell with $r/l = 0.03$. Positive compressive stress and strain magnitudes are used. The softening-onset strain $\varepsilon_{\text{onset}}$ is defined as the first strain at which the effective engineering stress falls 10% below the initial linear extrapolation $\sigma^* = E_0^* \varepsilon$. The quantities ε_{max} and σ^*_{max} denote the strain and stress at the maximum converged point of the plotted curve.

Model	$\varepsilon_{\text{onset}}$	σ^*_{onset}	ε_{max}	σ^*_{max}
Solid circular joints	0.0278	0.9979	0.0341	1.0079
Solid square joints	0.0273	0.9154	0.0307	0.9237
Hybrid circular joints	0.0322	1.2234	0.0400	1.2504
Hybrid square joints	0.0288	1.1047	0.0650	1.1651
Beam Timoshenko	0.0485	1.7561	0.0495	1.7625
Beam EB	0.0434	1.5710	0.0500	1.7030

The softening-onset results reinforce the interpretation of Figure 5.10(a). The solid element models soften at $\varepsilon_{\text{onset}} = 0.0278$ for circular joints and 0.0273 for square joints. The corresponding hybrid values are 0.0322 and 0.0288, while the beam-only models soften much later. The hybrid model therefore captures the onset of nonlinear softening more closely than the beam-only idealization, particularly for the square-joint case. The circular-joint hybrid model still delays the softening onset relative to the solid reference, but remains much closer to the solid response than either beam-only model.

The post-buckling response also shows a clear joint-geometry dependence. In the octet $r/l = 0.03$ case, the circular-joint models retain a higher effective stress than the square-joint models after softening. A plausible explanation is that the square joint geometry creates a locally stiffer joint–strut transition, which shifts curvature and deformation into the adjacent strut region and promotes earlier localization. The circular joint geometry appears to distribute this transition more smoothly, delaying localization and retaining a higher post-buckling load level. Since this explanation is based on the observed deformation response rather than a separate stress- or curvature-localization study, it should be interpreted as a qualitative mechanism.

For the purposes of the later joint-geometry studies, the most important result is not only the absolute agreement with the solid element reference, but also whether the model preserves the influence of joint geometry. Beam-only models collapse the joint region into a zero-volume node and therefore cannot directly represent the difference between circular and square joint transitions. The hybrid model retains these geometric differences in the solid joint subdomains, which allows it to reproduce joint-geometry trends more consistently than the beam-only idealization.

5.2.3. Observed Computational Cost Reduction

The accuracy improvements of the hybrid model are meaningful only if they are obtained at substantially lower computational cost than the corresponding solid element simulations. Table 5.12 therefore compares the degree-of-freedom and peak-memory reductions of the hybrid models relative to their solid element references. The table also reports the signed hybrid error in E_0^* and in effective elastic energy absorption, where the latter is available. Positive error indicates that the hybrid model overpredicts the corresponding solid element value, while negative error indicates underprediction. It is defined as

$$\delta_Q = 100\frac{Q_{\text{model}} - Q_{\text{solid}}}{Q_{\text{solid}}}.$$

where Q is the quantity of interest, which is either E_0^* or the effective elastic energy absorption. The same error metric is also applied to the beam-only models for comparison.

Table 5.12: Computational cost reduction and signed accuracy errors relative to the corresponding solid element reference. The degree-of-freedom and peak-memory reductions are computed from the solid and hybrid simulations of the same architecture, r/l, and joint geometry. Accuracy errors are reported for the hybrid, Timoshenko beam, and Euler–Bernoulli beam models. Positive error indicates overprediction relative to the corresponding solid element value, while negative error indicates underprediction. Peak-memory reductions are reported only where simulation-log data were available. The effective elastic energy absorption error is omitted for the octet $r/l = 0.03$ cases because those curves are treated using the softening-onset metric in Table 5.11.

Case	Joint	DOF red. [%]	Mem. red. [%]	Hybrid err.		Tim. err.		EB err.	
				E_0^* [%]	Energy [%]	E_0^* [%]	Energy [%]	E_0^* [%]	Energy [%]
Octet, $r/l = 0.03$	Circular	93.6	–	6.1	–	0.8	–	0.7	–
Octet, $r/l = 0.03$	Square	87.7	–	14.4	–	7.9	–	7.8	–
Octet, $r/l = 0.10$	Circular	73.9	94.6	9.5	8.8	-16.3	-16.4	-15.9	-18.4
Octet, $r/l = 0.10$	Square	59.4	94.5	13.5	12.9	-16.4	-16.3	-15.9	-18.4
Pyramidal, $r/l = 0.03$	Circular	91.2	99.3	-14.4	-13.3	-31.6	-29.2	-31.1	-31.7
Pyramidal, $r/l = 0.03$	Square	88.2	99.2	-17.0	-18.2	-36.2	-34.1	-35.7	-36.4
Pyramidal, $r/l = 0.10$	Circular	53.5	94.6	16.1	16.7	-49.4	-48.5	-45.0	-46.0
Pyramidal, $r/l = 0.10$	Square	58.5	96.4	16.5	16.8	-50.3	-49.4	-45.9	-47.0

Across the studied cases, the hybrid formulation reduces the number of degrees of freedom by approximately 54–94%. For the cases where memory data were available, the peak-memory reduction is approximately 95–99%. These reductions are obtained while retaining the solid description of the joint regions, which are the primary geometric feature of interest in this thesis.

The accuracy columns in Table 5.12 show that the hybrid model should be interpreted relative to the beam-only alternatives rather than only relative to the solid element reference. In the thick octet and thick pyramidal cases, the hybrid model overpredicts the solid response, while both beam-only models underpredict it. This is especially clear for the pyramidal $r/l = 0.10$ case, where the beam-only errors are approximately 45–50%, compared with hybrid errors of approximately 16–17%. In the $r/l = 0.03$ pyramidal case, all reduced models underpredict the solid reference, but the hybrid model remains substantially closer than either beam-only formulation. Part of this offset may originate from the non-identical boundary-condition implementation, particularly because the hybrid model constrains rotational interface degrees of freedom explicitly, whereas the solid element reference can only approximate the same constraint through translational restrictions on boundary face patches.

The magnitude of these errors should be interpreted differently from the small stiffness errors reported in linear reduced-order studies such as Portela *et al.* [12]. In that work, the reduced and fully resolved models were constructed to represent a closely matched linear interface problem, with the solid joint response condensed into an equivalent reduced description. In the present work, the comparison involves geometrically nonlinear compression, rigid-face beam–solid coupling, pure beam idealizations, and boundary conditions that cannot be imposed identically across all three model classes. The accuracy values in Table 5.12 therefore measure practical model-equivalence error under approximately matched nonlinear compression, rather than the isolated condensation error of a linear reduced-order node model.

A further distinction is that the validation evidence reported by Portela *et al.* [12] is concentrated on a linear octahedron unit-cell case over $r/l = 0.07$–0.15. That case is stretching-dominated, and the reported error is lowest for the most slender geometry (at 1%) before increasing with r/l up to 6%. The present results show a consistent trend: the best initial-stiffness agreement is obtained for the most slender stretching-dominated case considered here, namely the octet unit cell with $r/l = 0.03$. However, this is also the regime in which pure beam models already reproduce E_0^* reasonably well. Initial-stiffness accuracy alone therefore does not demonstrate the added value of the hybrid discretization.

All hybrid simulations reported in Table 5.12 were run on 10 shared-memory CPU cores. The memory values should therefore be interpreted as implementation-level measurements rather than hardware-independent complexity estimates. Nevertheless, the reductions are large enough to demonstrate the practical motivation for the hybrid approach: it enables joint-sensitive simulations at a fraction of the computational cost of fully solid unit-cell models.

For the joint-geometry studies developed later in this thesis, the relative response of circular and square joint geometries is as important as the absolute error relative to the solid element reference. To assess whether the hybrid model preserves this effect, the circular-to-square response ratio is defined as

$$\eta_Q = \frac{Q_{\text{circ}}}{Q_{\text{sq}}}, \tag{5.1}$$

where Q denotes either the initial effective Young's modulus, E_0^*, or the effective elastic energy absorption. A value of $\eta_Q > 1$ indicates that the circular-joint model gives a larger response than the square-joint model, while $\eta_Q < 1$ indicates the opposite. This ratio is useful because it measures whether the reduced model preserves the trend caused by changing joint geometry, independently of any common overprediction or underprediction of the absolute response.

Table 5.13: Circular-to-square response ratios for the solid and hybrid models. The ratio is defined as $\eta_Q = Q_{\text{circ}}/Q_{\text{sq}}$, where Q is either the initial effective Young's modulus, E_0^*, or the effective elastic energy absorption. Values close to one indicate a small joint-geometry effect, while agreement between the solid and hybrid ratios indicates that the hybrid model preserves the relative circular-to-square trend.

Case	$\eta_{E_0^*}^{\text{solid}}$	$\eta_{E_0^*}^{\text{hybrid}}$	$\eta_{\text{energy}}^{\text{solid}}$	$\eta_{\text{energy}}^{\text{hybrid}}$
Octet, $r/l = 0.03$	1.071	0.993	–	–
Octet, $r/l = 0.10$	1.000	0.964	1.000	0.964
Pyramidal, $r/l = 0.03$	0.933	0.963	0.932	0.987
Pyramidal, $r/l = 0.10$	0.983	0.980	0.983	0.981

Table 5.13 shows that the hybrid model captures the circular-to-square trend most clearly in the

pyramidal $r/l = 0.10$ case. For this bending-dominated thick-strut case, the solid and hybrid models give nearly identical ratios. For E_0^*, the solid ratio is approximately 0.983, while the hybrid ratio is approximately 0.980. For effective elastic energy absorption, the corresponding ratios are approximately 0.983 and 0.981. This indicates that, although the hybrid model overpredicts the absolute response in this case, it preserves the relative influence of circular and square joint geometry very well.

The agreement of the circular-to-square ratio is less uniform in the thinner cases. For the pyramidal $r/l = 0.03$ case, the solid model predicts a larger response for the square-joint geometry than for the circular-joint geometry, and the hybrid model preserves this ordering. However, the hybrid model predicts a smaller difference between the two joint geometries, especially for effective elastic energy absorption. This suggests that the hybrid model still captures the direction of the joint-geometry effect in this case, but underestimates its magnitude.

The octet cases require more careful interpretation. For the octet $r/l = 0.10$ case, the solid element reference predicts almost no difference between circular and square joints in either E_0^* or effective elastic energy absorption, while the hybrid model predicts a modestly larger response for the square-joint case. For the octet $r/l = 0.03$ case, the initial-modulus ratio is not preserved as well: the solid model predicts a higher E_0^* for circular joints, whereas the hybrid model gives nearly equal initial moduli for the two joint geometries. However, this case is dominated by geometric softening and post-buckling behavior rather than by the initial slope alone, so the circular-to-square comparison should be interpreted together with the softening-onset data in Table 5.11.

Overall, the circular-to-square ratio supports the main interpretation of the accuracy study. The hybrid model is not uniformly exact in absolute stiffness or energy absorption, but it can preserve the relative effect of joint geometry in the cases where finite joint geometry is mechanically important. This is especially clear for the pyramidal $r/l = 0.10$ case, which is also the case where beam-only models show the largest loss of accuracy. The ratio comparison therefore strengthens the argument that the hybrid formulation is useful for comparative joint-geometry studies, even when a residual absolute offset relative to the solid element reference remains.

5.2.4. Conclusions and Observations on Modeling Accuracy

The accuracy study shows that the hybrid element model is most useful in regimes where pure beam idealization removes mechanically important joint geometry. In slender stretching-dominated cases, especially the octet unit cell with $r/l = 0.03$, the beam-only models reproduce the initial effective Young's modulus reasonably well because the struts have high aspect ratio and the early response is close to the assumptions of beam theory. However, initial-stiffness accuracy alone does not demonstrate nonlinear accuracy. In the same octet case, the beam-only models delay the apparent onset of softening and overpredict the post-buckling stress, whereas the hybrid models remain closer to the solid element references.

This observation clarifies the main benefit of the hybrid discretization in slender stretching-dominated lattices. In the initial linear regime, finite joint geometry has only a limited influence on the effective stiffness, so the added value of the hybrid model becomes most evident once the response enters the geometrically nonlinear regime. For the octet $r/l = 0.03$ case, the hybrid formulation extends the usefulness of the Portela-type solid-joint/beam-strut discretization [12] beyond linear stiffness prediction into a regime where joint geometry affects nonlinear stability and post-buckling response.

The post-buckling response also helps explain why the hybrid model is not exact even in the slender stretching-dominated case. Before instability, the octet response is governed primarily by axial load transfer through slender struts. After local buckling, however, part of the deformation is carried by struts undergoing large rotations and bending-dominated deformation. The local mechanics of these buckled struts therefore become closer to the bending-dominated regimes where the hybrid model shows larger absolute errors. This suggests that part of the remaining post-buckling discrepancy may arise because the deformation mechanism changes after instability, even though the initial architecture is stretching-dominated.

The bending-dominated pyramidal cases show that the accuracy of the hybrid discretization does not transfer uniformly across all architectures. In the pyramidal $r/l = 0.10$ case, pure beam models strongly underpredict the solid response, while the hybrid models recover a substantial part of the missing stiffness and energy absorption. At the same time, the hybrid models overpredict the solid references by approximately 16–17%. This indicates that the solid-joint/beam-strut discretization remains useful in joint-sensitive bending-dominated cases, but its absolute accuracy is less controlled than in the slender stretching-dominated case. Part of this discrepancy may be associated with the stronger role of boundary-condition mismatch, rigid-face constraint assumptions, and bending-dominated deformation modes.

This interpretation should be viewed alongside the validation evidence reported by Portela *et al.* [12]. Their linear validation case is stretching-dominated, with errors ranging from approximately 1% for the most slender geometry to approximately 6% at larger r/l. Those results show that the solid-joint/beam-strut discretization can be highly accurate when the reduced and fully resolved models are closely matched linear problems. The larger errors reported here should therefore be interpreted in the context of the more demanding comparison considered in this thesis: geometrically nonlinear compression, post-buckling response, rigid-face coupling, and approximately matched boundary conditions.

The results also show that the hybrid model should not be described as a uniformly conservative or uniformly more compliant approximation. Depending on architecture and r/l, it may overpredict or underpredict the solid element reference. This behavior is expected because the comparison combines differences in structural idealization, finite joint representation, beam kinematics, rigid-face coupling, and boundary-condition implementation. The solid, hybrid, and beam-only models are therefore closely related compression tests rather than perfectly identical kinematic problems.

The most important conclusion is that the hybrid model preserves the geometric distinction between circular and square joints while reducing the computational cost by one to two orders of magnitude in the cases studied. It is therefore a suitable modeling approach for studying joint-geometry effects across nonlinear deformation regimes, provided that its remaining case-dependent error relative to solid element references is kept in view. The hybrid model should consequently be interpreted not as a complete replacement for solid element validation in its present form, but as a computationally efficient joint-sensitive model that is sufficiently informative for the comparative physics studies performed in this work. At the same time, the lower errors reported for closely matched linear reduced-order models in the literature suggest that more exact boundary-condition matching and validation on larger representative lattice samples could further reduce the observed error. Such validation is recommended before the method is used as a quantitative replacement for fully solid element models in design-critical applications.

6

Conclusion

This thesis developed and assessed a nonlinear-compatible hybrid beam–3D continuum modeling framework for architected truss metamaterials with explicit attention to joint geometry. The motivation for this work was the modeling gap between pure beam models and fully resolved solid finite element models. Beam models are computationally efficient and suitable for large lattice specimens, but they idealize the finite joint regions where important geometric effects may originate. Fully solid models resolve these regions directly, but their computational cost becomes prohibitive for low-relative-density lattices, especially when finite specimens, nonlinear deformation, and mesh refinement are required. The central objective of this thesis was therefore to investigate whether a hybrid formulation can provide a useful intermediate strategy: retaining three-dimensional fidelity in the joints while representing the connecting struts with beam elements.

The developed formulation treats the lattice as a coupled system of solid-joint and beam-strut submodels. Each beam–solid connection is represented by a six-degree-of-freedom interface state containing translations and rotations. On the solid side, this state is imposed through a rigid-face kinematic description; on the beam side, it corresponds to the beam-end motion. Coupling is enforced by requiring equilibrium between the force and moment resultants returned by the solid and beam submodels. This leads to a nonlinear interface residual problem, solved through a staggered partitioned strategy. The formulation preserves a clear separation between beam and solid solvers and therefore provides a modular computational structure that can, in principle, be extended to more advanced joint models, surrogate descriptions, or elastoplastic material behavior.

A significant part of the thesis was devoted to the solution strategy for this nonlinear interface problem. Several residual-based and Newton-type methods were considered, including fixed-point iteration with Aitken relaxation, Anderson acceleration, preconditioned residual descent, full-Jacobian Newton methods, and Jacobian-free Newton–Krylov methods. The final solver design combines preconditioned residual descent with Jacobian-free Newton–Krylov acceleration. In this strategy, preconditioned residual descent provides robust initial residual reduction and distributes the interface imbalance through the coupled lattice, while the Newton–Krylov stage provides faster local convergence once the interface state is sufficiently informative. This two-stage strategy proved important for making the staggered hybrid formulation practical for the nonlinear lattice cases considered in this work.

The computational cost analysis confirms the main motivation for the hybrid approach. In fully solid nonlinear lattice simulations, the total number of solid degrees of freedom remains the dominant analytical cost driver because it affects assembly, nonlinear iterations, linear solves, memory usage, and mesh refinement requirements. The hybrid formulation reduces this burden by replacing most of the strut volume with beam elements and restricting solid discretization to the joint regions. Across the accuracy cases studied in this thesis, the hybrid models reduced the degree-of-freedom count by approximately 54–94% relative to the corresponding solid element references. Where simulation-log data were available, peak-memory reductions were approximately 95–99%. These reductions demonstrate that the hybrid formulation can substantially lower the computational size of the problem while retaining explicit joint geometry.

At the same time, the cost study also showed that the hybrid model does not remove computational cost altogether; it changes where that cost is concentrated. The repeated local solid-joint solves are the dominant cost driver in the partitioned formulation. Refining the joint mesh increased wall time strongly even when the size of the global interface vector remained unchanged. Parallel execution of independent joint and beam submodels can reduce wall time, but the attainable speedup is bounded by the number of available tasks, load imbalance, and the serial outer coupling iterations. Increasing the number of unit cells increases both stored model state and total computational work, although in the small-scale tests performed here, interface-vector growth alone did not appear to dominate wall time. The practical implication is that future efficiency improvements should focus not only on interface solvers, but also on joint model resolution, local solver performance, and possible surrogate or reduced-order replacement of repeated joint solves.

The validation and accuracy studies show that the hybrid formulation is most useful in regimes where pure beam idealization removes mechanically important joint information. For slender stretching-dominated octet lattices, the beam-only models were able to reproduce the initial effective stiffness reasonably well, but they delayed the apparent onset of nonlinear softening and overpredicted the post-buckling stress. In these cases, the hybrid model remained closer to the solid reference in the nonlinear response because it retained finite joint geometry. For bending-dominated pyramidal lattices, the hybrid model recovered a substantial part of the stiffness and energy absorption that was missing from the pure beam models. However, its absolute agreement with the solid reference was less controlled and could include overprediction. The hybrid formulation should therefore not be interpreted as a uniformly conservative or uniformly exact replacement for full solid finite element modeling.

The main conclusion is that the hybrid beam–3D continuum formulation provides a useful joint-sensitive intermediate modeling strategy for comparative studies of architected truss metamaterials. Its value lies in combining explicit representation of joint geometry with a substantially reduced computational model size. It is particularly useful for identifying trends, comparing joint geometries, and studying cases where pure beam models are too idealized but fully solid models are too expensive for systematic investigation. At the same time, the remaining modeling errors are case-dependent and must remain visible. Full solid reference models are still necessary for case-specific validation, especially in strongly nonlinear, instability-dominated, or quantitatively critical regimes.

Within these limits, the thesis demonstrates that nonlinear-compatible hybrid modeling is a promising direction for studying joint-geometry effects in low-relative-density truss lattices. The framework developed here establishes the formulation, solver structure, validation procedure, and a first accuracy assessment for that purpose. It provides a practical basis for future work on improved interface coupling, adaptive joint resolution, surrogate joint models, larger finite specimens, and material nonlinearities. Most importantly, it shows that joint geometry can be retained as an explicit modeling variable without requiring the entire lattice to be discretized as a fully three-dimensional continuum model.

7

Recommendations for Future Work

The hybrid beam–3D continuum formulation developed in this thesis provides a promising intermediate modeling strategy for joint-sensitive lattice simulations, but several aspects should be developed further before the method can be used as a general predictive tool. The most important recommendations concern solver formulation, accuracy assessment, nonlinear material behavior, and validation against experiments.

A first recommendation is to investigate a monolithic implementation of the hybrid beam–3D continuum formulation. The staggered formulation used in this thesis preserves modularity and allows the beam and solid solvers to remain separate, but it also leads to a nonlinear interface-equilibrium problem that requires specialized residual-based and Newton–Krylov solution strategies. A monolithic formulation would assemble the beam, solid, and interface equations into one coupled nonlinear system. This may reduce the need for complex outer interface iterations and could improve robustness in strongly nonlinear regimes. The main challenge is implementation complexity, since consistent tangent contributions and interface coupling terms must be assembled within a unified solver framework. Nevertheless, such a formulation would provide an important benchmark for judging whether the observed computational cost is mainly caused by the hybrid discretization itself or by the staggered coupling strategy.

A second recommendation is to further optimize the staggered interface solver. If the partitioned framework is retained, future work should investigate more systematic preconditioner construction, adaptive switching between residual descent and Newton–Krylov stages, adaptive load-step control, preconditioner reuse between nearby load steps, and improved line-search criteria. The present work showed that the number of residual evaluations can strongly influence wall time, while each residual evaluation requires repeated local joint and beam solves. Reducing the number of failed or weakly productive residual evaluations is therefore a direct pathway to improving computational efficiency. More advanced connectivity-based preconditioners, inexact Newton forcing terms, and better parallel scheduling of independent joint solves are especially relevant directions.

A third recommendation is to explore surrogate or reduced-order descriptions of nonlinear joint response within the staggered framework. The cost study showed that repeated local solid-joint solves are a dominant computational cost driver. This makes the joint submodel a natural target for surrogate replacement. A future framework could train or construct nonlinear joint operators that map rigid-face interface motions to force and moment resultants, while retaining the same interface residual structure used in this thesis. Such surrogates would need careful validation over the relevant deformation range, especially near buckling, yielding, or other nonlinear transitions. If successful, this would preserve the modular advantage of the staggered hybrid formulation while substantially reducing the cost of repeated joint evaluations.

A fourth recommendation is to repeat the accuracy study with more closely matched boundary conditions between the beam, hybrid, and solid models. In the present work, the comparison between model classes is affected by unavoidable differences in how boundary conditions are imposed on beam ends, rigid faces, and fully resolved solid surfaces. This means that the reported accuracy values measure practical model-equivalence error under approximately matched compression, rather than the isolated error of the hybrid discretization alone. A future study should design benchmark cases in which boundary conditions, loading definitions, strain measures, and reaction-force extraction are matched as closely as possible across all model classes. This would allow more precise error metrics for stiffness, softening onset, post-buckling stress, and energy absorption.

A fifth recommendation is to extend the material modeling beyond elastic geometric nonlinearity. The present thesis focuses on materially elastic lattices, so the nonlinear response arises from large deformation, rotation, instability, and load-path redistribution. However, many experimentally relevant additively manufactured lattices undergo plasticity, damage, fracture, and contact during compression. Implementing elastoplastic material models in the solid joint regions would be the most important next step, because it would allow the hybrid formulation to be compared against real large-deformation compression experiments on printed lattices. Further extensions could include fracture, contact between collapsing members, viscoelasticity, or damage models, although these should be introduced only after the elastoplastic case has been validated.

A sixth recommendation is to investigate alternative interface coupling strategies. The rigid-face MPC used in this thesis provides a simple and robust six-degree-of-freedom coupling between beam ends and solid faces, but it also suppresses face warping and higher-order deformation at the interface. This may contribute to some of the observed modeling error. Future work should compare the present rigid-face coupling with averaged MPC or RBE3-type coupling, mortar-type transfer, penalty methods, Nitsche-type coupling, or other weak interface formulations. Such a comparison would help determine how much of the accuracy limitation lies in the hybrid beam–solid concept itself or in the specific rigid-face kinematic assumption used at the interface.

Finally, the method should be tested on a broader set of lattice architectures, joint geometries, and specimen sizes. The present work focuses on selected circular and square joint geometries and a limited set of topology and slenderness cases. Future studies should include additional joint modifications such as fillets, rounded nodes, star-like sections, and density-preserving shape transitions. Larger finite specimens should also be considered to assess whether the conclusions remain valid when localization, boundary effects, and layer-wise collapse become more pronounced. Combined with experimental validation, this broader parameter study would clarify where the hybrid beam–3D continuum formulation is most reliable and where full solid modeling remains necessary.

Bibliography

[1] R. Lakes. “Foam Structures with a Negative Poisson’s Ratio”. *Science* 235 (4792 1987), pp. 1038–1040.

[2] A. G. Evans, J. W. Hutchinson, N. A. Fleck, M. F. Ashby, and H. N. G. Wadley. “The topological design of multifunctional cellular metals”. *Progress in Materials Science* 46.3–4 (2001), pp. 309–327. DOI: 10.1016/S0079-6425(00)00016-5.

[3] H.N.G. Wadley. “Multifunctional periodic cellular metals”. *Philosophical Transactions of the Royal Society A* 364 (2006), pp. 31–68. DOI: 10.1098/rsta.2005.1697.

[4] N. A. Fleck, V. S. Deshpande, and M. F. Ashby. “Micro-architectured materials: past, present and future”. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 466.2121 (2010), pp. 2495–2516. DOI: 10.1098/rspa.2010.0215.

[5] 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.

[6] T. A. Schaedler and W. B. Carter. “Architected Cellular Materials”. *Annual Review of Materials Research* 46 (July 2016), pp. 187–210. DOI: 10.1146/annurev-matsci-070115-031624.

[7] K. Bertoldi, V. Vitelli, J. Christensen, and M. van Hecke. “Flexible mechanical metamaterials”. *Nature Reviews Materials* 2 (2017), p. 17066. DOI: 10.1038/natrevmats.2017.66.

[8] L. Gibson and M. Ashby. *Cellular Solids: Structure and Properties, Second Edition*. Cambridge, U.K.: Cambridge University Press, 1997.

[9] M. Benedetti, A. du Plessis, R. O. Ritchie, M. Dallago, S. M. J. Razavi, and F. Berto. “Architected cellular materials: A review on their mechanical properties towards fatigue-tolerant design and fabrication”. *Materials Science and Engineering: R: Reports* 144 (2021), p. 100606. DOI: 10.1016/j.mser.2021.100606.

[10] V. Deshpande, M. Ashby, and N. Fleck. “Foam Topology Bending Versus Stretching Dominated Architectures”. *Acta Materialia* 49 (2001), pp. 1035–1040. DOI: 10.1016/S1359-6454(00)00379-7.

[11] R. Hutchinson and N. Fleck. “The structural performance of the periodic truss”. *Journal of the Mechanics and Physics of Solids* 54.4 (2006), pp. 756–782. DOI: https://doi.org/10.1016/j.jmps.2005.10.008.

[12] C. M. Portela, J. R. Greer, and D. M. Kochmann. “Impact of node geometry on the effective stiffness of non-slender three-dimensional truss lattice architectures”. *Extreme Mechanics Letters* 22 (July 2018), pp. 138–148. DOI: 10.1016/J.EML.2018.06.004.

[13] X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, S. O. Kucheyev, N. X. Fang, and C. M. Spadaccini. “Ultralight, ultrastiff mechanical metamaterials”. *Science* 344.6190 (2014), pp. 1373–1377. DOI: 10.1126/science.1252291. eprint: https://www.science.org/doi/pdf/10.1126/science.1252291.

[14] L. R. Meza, S. Das, and J. R. Greer. “Strong, lightweight, and recoverable three-dimensional ceramic nanolattices”. *Science* (2014). DOI: 10.1126/science.1255908.

[15] L. Valdevit, A. J. Jacobsen, J. R. Greer, and W. B. Carter. “Protocols for the Optimal Design of Multi-Functional Cellular Structures: From Hypersonics to Micro-Architected Materials”. *Journal of the American Ceramic Society* 94.s1 (2011), S15–S34. DOI: 10.1111/j.1551-2916.2011.04599.x.

[16] 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: 10.1002/adem.201800864.

[17] R. Glaesener, S. Kumar, C. Lestringant, T. Butruille, C. Portela, and D. Kochmann. “Predicting the Influence of Geometric Imperfections on the Mechanical Response of 2D and 3D Periodic Trusses”. *Acta Materialia* 254 (2023), p. 118918. DOI: 10.1016/j.actamat.2023.118918.

[18] R. M. Latture, R. X. Rodriguez, L. R. Holmes, and F. W. Zok. “Effects of nodal fillets and external boundaries on compressive response of an octet truss”. *Acta Materialia* 149 (2018), pp. 78–87. DOI: https://doi.org/10.1016/j.actamat.2017.12.060.

[19] P.-T. Doutre, C. Grandvallet, L. Gobet, F. Vignat, and R. Dendievel. “Influence of fillets onto mechanical properties of octet-truss lattice structures”. *The International Journal of Advanced Manufacturing Technology* 132.5–6 (2024), pp. 2503–2516. DOI: 10.1007/s00170-024-13512-z.

[20] J. C. Maxwell. “On the calculation of the equilibrium and stiffness of frames”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 27.182 (1864), pp. 294–299. DOI: 10.1080/14786446408643668.

[21] Z. Sun, Y. Guo, and V. Shim. “Deformation and energy absorption characteristics of additively-manufactured polymeric lattice structures — Effects of cell topology and material anisotropy”. *Thin-Walled Structures* 169 (Dec. 2021), p. 108420. DOI: 10.1016/j.tws.2021.108420.

[22] 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: 10.1016/J.COMPSTRUCT.2022.116397.

[23] J. Bauer, L. Meza, T. Schaedler, R. Schwaiger, X. Zheng, and L. Valdevit. “Nanolattices: An Emerging Class of Mechanical Metamaterials”. *Advanced Materials* 29 (Sept. 2017), p. 1701850. DOI: 10.1002/adma.201701850.

[24] M. Casata, C. Garrido, T. Wilkinson, S. Perosanz, and D. Barba Cancho. “Nodal geometry optimization in additively manufactured thin lattice structures: Application to the design of lightweight Ti-6Al-4V LPBF metamaterials”. *Thin-Walled Structures* 221 (Dec. 2025), p. 114414. DOI: 10.1016/j.tws.2025.114414.

[25] G. Dong and Y. F. Zhao. “Numerical and experimental investigation of the joint stiffness in lattice structures fabricated by additive manufacturing”. *International Journal of Mechanical Sciences* 148 (2018), pp. 475–485. DOI: `https://doi.org/10.1016/j.ijmecsci.2018.09.014`.

[26] C. Portela, G. Phlipot, J. Greer, and D. Kochmann. “Approaches for Efficient Modeling of the Mechanical Response of Complex Periodic Truss Architectures”. Apr. 2019. DOI: `10.33599/nasampe/s.19.1428`.

[27] T. De Weer, N. Vannieuwenhoven, N. Lammens, and K. Meerbergen. “The parametrized superelement approach for lattice joint modelling and simulation”. *Computational Mechanics* 70.2 (2022), pp. 451–475. DOI: `10.1007/s00466-022-02176-9`.

[28] Bianca Giovanardi’s research group. *TU Delft’s Gitlab page for Summit lite software*. Online: https://gitlab.tudelft.nl/aces/summit-lite. 2022.

[29] M. Zhu, F. McKenna, and M. H. Scott. “OpenSeesPy: Python library for the OpenSees finite element framework”. *SoftwareX* 7 (2018), pp. 6–11. DOI: `10.1016/j.softx.2017.10.009`.

[30] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. 2nd ed. Chichester: John Wiley & Sons, 2014.

[31] S. C. Klarmann, J. Wackerfuß, and S. Klinkel. “Coupling 2D Continuum and Beam Elements: A Mixed Formulation for Avoiding Spurious Stresses”. *Computational Mechanics* 70.6 (2022), pp. 1145–1166. DOI: `10.1007/s00466-022-02221-7`.

[32] P. Wriggers. *Computational Contact Mechanics*. 2nd ed. Berlin, Heidelberg: Springer, 2006. DOI: `10.1007/978-3-540-32609-0`.

[33] I. Babuška. “The Finite Element Method with Lagrangian Multipliers”. *Numerische Mathematik* 20 (1973), pp. 179–192. DOI: `10.1007/BF01436561`.

[34] J. Nitsche. “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36 (1971), pp. 9–15. DOI: `10.1007/BF02995904`.

[35] P. Hansbo and J. Hermansson. “Nitsche’s Method for Coupling Non-Matching Meshes in Fluid-Structure Vibration Problems”. *Computational Mechanics* 32.1–2 (2003), pp. 134–139. DOI: `10.1007/s00466-003-0467-7`.

[36] F. Ben Belgacem, P. Hild, and P. Laborde. “The Mortar Finite Element Method for Contact Problems”. *Mathematical and Computer Modelling* 28.4–8 (1998), pp. 263–271. DOI: `10.1016/S0895-7177(98)00121-6`.

[37] NASA. *The NASTRAN User’s Manual*. NASA Special Publication NASA-SP-222(06). National Aeronautics and Space Administration, 1983.

[38] U. Küttler and W. A. Wall. “Fixed-point fluid–structure interaction solvers with dynamic relaxation”. *Computational Mechanics* 43.1 (2008), pp. 61–72. DOI: `10.1007/s00466-008-0255-5`.

[39] D. G. Anderson. “Iterative procedures for nonlinear integral equations”. *Journal of the ACM* 12.4 (1965), pp. 547–560. DOI: `10.1145/321296.321305`.

[40] H. F. Walker and P. Ni. “Anderson acceleration for fixed-point iterations”. *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735. DOI: `10.1137/10078356X`.

[41] D. A. Knoll and D. E. Keyes. “Jacobian-free Newton–Krylov methods: a survey of approaches and applications”. *Journal of Computational Physics* 193.2 (2004), pp. 357–397. DOI: `10.1016/j.jcp.2003.08.010`.

[42] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia: SIAM, 1995. DOI: `10.1137/1.9781611970944`.

[43] Y. Saad and M. H. Schultz. “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”. *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869. DOI: `10.1137/0907058`.

[44] S. C. Eisenstat and H. F. Walker. “Choosing the forcing terms in an inexact Newton method”. *SIAM Journal on Scientific Computing* 17.1 (1996), pp. 16–32. DOI: `10.1137/0917003`.

[45] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Vol. 2. Fundamentals of Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006. DOI: `10.1137/1.9780898718881`.

[46] A. George. “Nested Dissection of a Regular Finite Element Mesh”. *SIAM Journal on Numerical Analysis* 10.2 (1973), pp. 345–363. DOI: `10.1137/0710032`.

[47] J. R. Gilbert and R. E. Tarjan. “The Analysis of a Nested Dissection Algorithm”. *Numerische Mathematik* 50.4 (1986), pp. 377–404. DOI: `10.1007/BF01396660`.

[48] R. J. Lipton, D. J. Rose, and R. E. Tarjan. “Generalized Nested Dissection”. *SIAM Journal on Numerical Analysis* 16.2 (1979), pp. 346–358. DOI: `10.1137/0716027`.

[49] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. “Geometric Separators for Finite-Element Meshes”. *SIAM Journal on Scientific Computing* 19.2 (1998), pp. 364–386. DOI: `10.1137/S1064827594262613`.

[50] A. Brandt. “Multi-Level Adaptive Solutions to Boundary-Value Problems”. *Mathematics of Computation* 31.138 (1977), pp. 333–390. DOI: `10.1090/S0025-5718-1977-0431719-X`.

[51] K. Stüben. “A Review of Algebraic Multigrid”. *Journal of Computational and Applied Mathematics* 128.1–2 (2001), pp. 281–309. DOI: `10.1016/S0377-0427(00)00516-1`.

[52] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge. “Robustness and Scalability of Algebraic Multigrid”. *SIAM Journal on Scientific Computing* 21.5 (2000), pp. 1886–1908. DOI: `10.1137/S1064827598339402`.

[53] M. Griebel, D. Oeltz, and M. A. Schweitzer. “An Algebraic Multigrid Method for Linear Elasticity”. *SIAM Journal on Scientific Computing* 25.2 (2003), pp. 385–407. DOI: `10.1137/S1064827502407810`.

[54] V. Mishra and K. Suresh. “Fast Iterative Solvers for Thin Structures”. *Finite Elements in Analysis and Design* 47.11 (2011), pp. 1223–1231. DOI: `10.1016/j.finel.2011.05.009`.