THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

École doctorale n°37

SPIM - Sciences Physiques pour l'Ingénieur et Microtechniques

Doctorat de **Mécanique**

Par

Cadart Thomas

Domain Decomposition Surrogate Framework for the Optimal Design of Locally Tailored Lattice Structures

Thèse soutenue à UTBM Sevenans, le 10 décembre 2025 devant le jury composé de :

DELAMÉZIÈRE, Arnaud	Professeur Institut Mines-Télécom	GIP-InSIC	Président
ELGUEDJ, Thomas	Professeur des universités	INSA Lyon	Rapporteur
NACEUR, Hakim	Professeur des universités	INSA Hauts-de-France	Rapporteur
LEBAAL, Nadhir	Professeur des universités	UTBM	Directeur de thèse
ROTH, Sébastien	Professeur des universités	UTBM	Co-directeur
DEMOLY, Frédéric	Professeur des universités	UTBM	Invité et encadrant
HIRSCHLER, Thibaut	Maître de Conférences	UTBM	Invité et encadrant
BAHI, Mohamed Slim	Ingénieur de recherche	Université de Lorraine	Invité et encadrant

Remerciements

Une quantité assez impressionnante de personnes m'ont permis d'en arriver là. Réussir à écrire ce document a nécessité un véritable alignement des étoiles, qui n'est pas donné à tout le monde.

J'ai eu la chance inouïe d'avoir des parents qui ont été présents pour moi tout au long de ma scolarité, et qui m'ont aidé à me dépasser et à surmonter les épreuves. Ensuite, ils m'ont encouragé à poursuivre mes études, en faisant des sacrifices afin que je sois dans les meilleures conditions possibles. Papa, Maman, Merci : c'est grâce à tout cela que j'en suis là où je suis aujourd'hui. Merci à mon frère, Ugo, qui m'a appris la combativité depuis mon plus jeune âge, ce qui m'a grandement servi ces dernières années (plus mentalement que physiquement cette fois-ci 😄).

Les rencontres sont des sources d'inspiration et de motivation, et elles ont été nombreuses tout au long de cette thèse. Merci à tous ces collègues devenus amis : Célestin pour tous les murs grimpés, Julien pour les nombreuses soirées jeux de société qui m'ont permis de souffler, Margaux pour la fraîcheur d'esprit que tu m'as inspirée, et Greg pour cette légèreté de vie qui donne envie d'être toujours à bloc.

Merci à tous mes amis qui se sont donné beaucoup de mal pour assister à ma soutenance ; cela restera gravé dans mon cœur : Pierrot, Céline et Julien. Merci également à toutes les personnes présentes en visio-conférence, probablement en double écran pendant le travail 😄.

Merci à tous les doctorants du bureau R414 pour la bonne ambiance : Hongchuan, Robin et Oualid. Merci au programmeur du laboratoire, Hichem, pour toutes ces heures passées à discuter sciences ; avec tout le travail et l'énergie que tu y mets, tu vas y arriver au doctorat.

Merci à toute l'équipe de DOCéo, à laquelle j'ai eu la chance d'appartenir, et qui accomplit un travail essentiel en faveur de la santé mentale des doctorants, un aspect fondamental dont il est indispensable de prendre soin.

Merci à mon équipe encadrante : Nadhir, qui m'a permis de gagner en autonomie ; Sébastien, toujours à se démener pour faire avancer les choses ; Frédéric, qui corrige les

articles à n'importe quelle heure ; Thibaut, qui m'a permis d'aller tellement plus loin et qui m'a montré comment être un bon scientifique ; et Slim, qui m'a révélé les failles insoupçonnées du fonctionnement de l'académie.

Merci à mes rapporteurs, qui ont donné de leur temps bénévolement, pourtant très limité, pour relire ce manuscrit : Elguedj Thomas et Naceur Hakim. Merci à mon président de jury pour ce très beau rapport de soutenance : Delamézière Arnaud.

Enfin, merci à ma plus belle rencontre durant cette thèse : Sarah, ma chérie, mon aventurière, mon boost, qui m'a permis de garder le moral. Sans tout cet amour que tu m'envoies chaque jour, ma vie serait bien plus morose. J'ai hâte de vivre encore de nombreuses aventures avec toi, et très prochainement au Canada.

Et pour conclure, mon dernier mot de cette thèse est destiné à toutes les personnes qui la liront : n'oubliez jamais que « Tout est lattice ».

Résumé

Les progrès récents des technologies de fabrication, et en particulier de la fabrication additive, ont ouvert de nouvelles perspectives pour la conception de structures complexes. Parmi elles, les structures lattices ont suscité un intérêt considérable, car elles permettent la création de structures périodiques présentant des propriétés et des fonctionnalités auparavant non réalisables. Cependant, la conception et l'optimisation de telles géométries complexes afin d'obtenir des comportements spécifiques demeurent particulièrement difficiles, principalement en raison du coût de calcul élevé inhérent aux chaînes numériques classiques. Pour analyser et comprendre efficacement les structures lattices, de nouvelles chaînes de calcul doivent donc être développées. Le premier chapitre de cette thèse s'est concentré sur la simulation de structures lattices de type treillis modélisées à l'aide d'éléments poutres. La principale difficulté de cette approche réside dans la représentation précise de la raideur des jonctions, un aspect particulièrement complexe à modéliser avec des éléments unidimensionnels. Une méthode généralisée d'ajustement de la raideur des jonctions a été développée en définissant une zone de pénalisation dépendante du rayon des poutres et des angles entre celles-ci. Cette stratégie améliore significativement la cohérence entre les simulations numériques et les résultats expérimentaux. Sur cette base, une approche améliorée couplant la décomposition de domaine et les modèles de substitution a été développée afin d'accroître l'efficacité numérique. Les structures lattices se prêtent naturellement à une décomposition cellule par cellule, ce qui permet à la méthode de décomposition de domaine de réduire considérablement le coût de calcul par rapport aux approches classiques. De plus, le calcul préliminaire du comportement mécanique de cellules unitaires paramétrées en fonction de leur géométrie, combiné à l'utilisation de modèles de substitution, permet de prédire efficacement la réponse mécanique globale de la structure. Deux stratégies distinctes de modèle de substitution ont été introduites : l'une fondée sur la réduction de modèle, l'autre sur les réseaux de neurones enrichis par la physique, chacune apprenant de manière différente la relation entre les paramètres géométriques et les réponses mécaniques. Cette approche se prête particulièrement bien aux cadres d'optimisation pour deux raisons principales : elle est numériquement efficace et, grâce au modèle de substitution, la sensibilité des paramètres de conception peut directement être obtenue. Enfin, un cadre d'optimisation intégrant l'ensemble des méthodes présentées a été développé et testé sur plusieurs cas de référence. L'algorithme d'optimisation basé sur le gradient a démontré sa robustesse, sa précision et son efficacité pratique, tant pour des problèmes de conception simples que de haute dimension. Dans l'ensemble, cette thèse présente une chaîne numérique unifiée et efficace, combinant une approche avancée de simulation à la flexibilité de conception offerte par les structures lattices.

Note : *ce document est intégralement écrit en anglais. Un résumé détaillé écrit en français est donné à la fin du document, en Chapitre 7.*

Abstract

Advances in manufacturing technologies, and in particular additive manufacturing, have opened new possibilities for designing highly complex structures. Among them, lattice structures have gained considerable attention as they enable the creation of material architectures exhibiting properties and functionalities that were previously unattainable. However, designing and optimizing such complex shapes to achieve specific behaviors remains highly challenging, mainly due to the significant computational burden inherent to classical numerical frameworks. To efficiently analyze and understand lattice structures, new computational frameworks must therefore be developed. The first chapter of this thesis focused on the simulation of truss-based lattice structures modeled using beam elements. The main difficulty associated with this approach lies in the accurate representation of joint stiffness, which is particularly challenging when using one-dimensional elements. A generalized joint stiffening method was developed by defining a penalization zone based on the beam radius and the angle between connected beams. This strategy significantly improves the consistency between numerical simulations and experimental results. Building on this work, an enhanced approach that couples domain decomposition with surrogate modeling techniques was developed to further improve computational efficiency and scalability. Lattice structures are naturally decomposed on a cell-by-cell basis, which allows the domain decomposition method to significantly reduce the computational burden compared to classical approaches. Moreover, precomputing the mechanical behavior of shape-parameterized unit cells and coupling them with surrogate models enables the efficient prediction of the global mechanical response. Two distinct surrogate modeling strategies were introduced: one based on reduced-order modeling and the other on physics-augmented neural networks, each capturing the relationship between geometric parameters and mechanical responses in a different manner. This approach is particularly well suited to optimization frameworks for two main reasons. It is computationally efficient, and thanks to the surrogate model, the sensitivity of the design parameters can be directly recovered. Finally, an optimization framework integrating all the previously presented methods was developed and tested on multiple benchmark problems. The gradient-based optimization algorithm demonstrated robustness, accuracy, and practical efficiency in both simple and high-dimensional design spaces. Experimental tests were conducted as a final step to assess the performance of the proposed framework. Overall, this thesis presents a unified and computationally efficient framework that couples an advanced simulation approach with the design flexibility of lattice structures.

Table of contents

Remerciements iii

Résumé v

Abstract vii

Table of contents xii

List of Acronyms xv

1 Introduction 1

1.1 Motivations and Objectives 1

1.2 Outline 3

1.3 Contributions 5

2 A Global Vision for Optimal Design of Lattice Structures 7

2.1 Lattice Structures Design Driving the Overall Performance 8

2.1.1 A Large Variety of Lattice Structures 9

2.1.2 Multiphysical Properties of Lattice Structures 12

2.1.3 Manufacturing Strategies and Constraints for Lattice Structures . 15

2.1.4 Opportunities and Challenges in Generative Design 17

2.1.5 Multidisciplinary Applications of Lattice Structures 19

2.2 Efficient Property Evaluation for Lattice Structures 21

2.2.1 Single-Scale Modeling Approaches 21

2.2.2 Multiscale and Decomposition Strategies 26

2.2.3 Experimental Characterization 31

2.3 Optimal Design for Mechanical Lattice Structures 34

2.3.1 Optimization Strategies at the Cell-Scale 35

2.3.2 Optimization Strategies at the Macro-Scale 37

2.3.3 Optimization Strategies across Multiple Scales 39
2.3.4 Metaheuristic Strategies 40
2.4 Accelerating Optimization with Surrogate Models 43
2.4.1 Classification of Surrogate Modeling Approaches 43
2.4.2 Levels of Surrogate Model Integration 53
2.4.3 Potential and Limitations of Surrogate Modeling 58
2.5 Scientific Challenges and Positioning 59

3 Enhanced Beam Modeling for Lattice Structures 61
3.1 Beam Models for Lattice Structures: An Efficient Yet Challenging Approach 62
3.1.1 Beam Model Formulation 62
3.1.2 Challenges in Modeling Joint Stiffness in Beam Models 64
3.1.3 Lattice Geometries, Material Properties, and Experimental Framework 67
3.2 Identification of Optimal Stiffening Parameters via Static Analyses 70
3.2.1 Assessment of the Beam Models Accuracy over Unit Cells 70
3.2.2 Defining the Length of the Stiffening Region 72
3.2.3 Defining the Penalty Coefficient 74
3.2.4 Evaluating the Effects of the Stiffening Region on Beam Model Accuracy 74
3.2.5 Optimizing the Penalty Coefficient 76
3.3 Application of the Joint Stiffening Method to Dynamic Simulations . . . 77
3.3.1 Assessment of the Beam Models Accuracy via Modal Analyses . . 78
3.3.2 Balance the Mass Distribution 80
3.3.3 Evaluating Added Mass Effects on the Structural Dynamics . . . 80
3.4 Extension to Nonlinear Analyses and Experimental Comparison 82
3.4.1 Uniaxial Compression of a Unit-cell Made of Elasto-Plastic Material 82
3.4.2 Experimental Validation on 3D-printed BCC Structures 84
3.5 Conclusion on the Beam Model Penalization Methodology 87

4 Surrogate-Enhanced Domain Decomposition for Efficient Simulation 91
4.1 Foundations for Exploring Locally Tailored Lattice Structures 92
4.1.1 Hybrid Lattice Geometry Paving a Large Design Space 93
4.1.2 Benchmark Structural Configurations 95
4.2 Domain Decomposition Method Applied to Lattice Structures 97
4.2.1 Global overview of DDM 98
4.2.2 Primal interface problem 102

4.2.3 Solving the Interface Problem . . . 104
4.2.4 Application to Beam-like Lattice Structures . . . 106
4.3 Surrogate Model for Fast Analysis . . . 110
4.3.1 Reduced Basis via Greedy Algorithm . . . 111
4.3.2 A variety of Surrogate Modeling Techniques . . . 115
4.3.3 Model Behavior and Accuracy Assessment . . . 119
4.4 Embedding Physical Knowledge into Neural Network . . . 126
4.4.1 Embedding Physics for Efficient and Generalizable Surrogate Models 127
4.4.2 Training Strategy for the PANN . . . 134
4.4.3 Efficient Dataset Generation . . . 137
4.4.4 Training Procedure and Evaluation of the PANN . . . 141
4.4.5 Nonlinear CG Solver for PANN-Based DDM . . . 145
4.5 Discussion and Conclusion . . . 148

5 Towards Efficient Optimization of Locally Tailored Lattice Structures 153
5.1 Formulation of the Optimization Framework . . . 154
5.1.1 Optimization Problem . . . 154
5.1.2 Surrogate Model for Relative Density . . . 157
5.2 Optimization Results and Design Space Analyses . . . 158
5.2.1 Effect of the Relative Density Constraint on Structural Performance 158
5.2.2 Post-Processing . . . 160
5.2.3 Three-point Bending Beam Study Case with Different Parameter Spaces . . . 161
5.2.4 Case Study: Full Optimization Pipeline Applied to a Femoral Head Implant . . . 165
5.3 Experimental Validation on Benchmark Structures . . . 168
5.3.1 Numerical Optimization Performance . . . 168
5.3.2 Fabrication, Post-Processing, and Experimental Preparation . . . 170
5.3.3 Experimental Results and Discussion . . . 173
5.4 Concluding Remarks on the Optimization of Locally Tailored Lattice Structures . . . 175

6 Conclusion And Perspectives 177
6.1 Final Overview . . . 177
6.2 Research Perspectives . . . 179
6.2.1 Short-term perspectives . . . 179
6.2.2 Medium-term perspectives . . . 180

6.2.3 Long-term perspectives 180

7 Résumé des travaux en Français **181**
7.1 Introduction 182
7.2 Rapide présentation de l'état de l'art et problématique 183
7.2.1 Structures lattices et fabrication additive 183
7.2.2 Simulation des structures lattices 184
7.2.3 Optimisation des structures lattices 185
7.2.4 Verrous scientifiques et positionnement de la démarche 186
7.3 Méthode de pénalisation aux nœuds 187
7.3.1 Contexte et motivation 187
7.3.2 Méthodologie 188
7.3.3 Extension aux cas non linéaires et validation expérimentale 194
7.3.4 Conclusion 194
7.4 Décomposition de domaine pour les structures lattices 197
7.4.1 Fondations pour l'exploration des lattices localement optimisées . 197
7.4.2 Méthode de décomposition de domaine appliquée aux structures lattices 197
7.4.3 Modèle de substitution pour l'analyse rapide 201
7.4.4 Modèle de substitution augmenté par la physique 204
7.4.5 Discussion et conclusion 206
7.5 Vers une optimisation locale efficace des structures lattices 206
7.5.1 Étude de cas : Prothèse de tête fémorale 207
7.5.2 Validation expérimentale sur les structures de référence 208
7.6 pyLattice : un outil de conception, de simulation et d'optimisation 210
7.7 Conclusion et perspectives 212

References **213**

Appendix A pyLattice: A design, simulation, and optimization tool for truss lattice structures **239**
A.1 Designing Lattice Structures with pyLattice 240
A.2 Simulation possibilities with pyLattice 242
A.3 Optimization possibilities with pyLattice 243
A.4 Pyrough compatibility 244

Appendix B Derivative RBF **247**

List of Acronyms

AD	Automatic Differentiation.
AE	Autoencoder.
AH	Asymptotic Homogenization.
AM	Additive Manufacturing.
AUX	Auxetic.
BCC	Body-Centered Cubic.
BCC+C	Reinforced Body-Centered Cubic with beam in horizontal plane at center node.
BCCZ	Body-Centered Cubic with beam on Z-axis.
BDD	Balanced Domain Decomposition.
CAD	Computer-Aided Design.
CFD	Computational Fluid Dynamics.
CG	Conjugate Gradient.
CNN	Convolutional Neural Network.
CT	Computed Tomography.
CutFEM	Cut Finite Element Method.
DDM	Domain Decomposition Method.
DIW	Direct Ink Writing.
DL	Deep Learning.
DLP	Digital Light Processing.
DOF	Degree of Freedom.
EBM	Electron Beam Melting.

F2FCCZ Face Centered Cubic with beam on Z-axis.
FA Fabrication Additive.
FE Finite Element.
FE^2 Two-scale Finite Element.
FEM Finite Element Method.
FETI Finite Element Tearing and Interconnecting.
FETI-DP FETI-Dual-Primal.
FFF Fused Filament Fabrication.

GA Genetic Algorithms.
GAC Gibson-Ashby Cell.
GAN Generative Adversarial Network.
GNN Graph Neural Network.
GPR Gaussian Process Regression.
GRP Gaussian Process Regression.

IBM Immersed Boundary Method.
IGA Isogeometric Analysis.

LHS Latin Hypercube Sampling.

MAE Mean Absolute Error.
MBB Messerschmitt–Bölkow–Blohm.
MDD Méthode de Décomposition de Domaine.
MSE Mean Squared Error.

NCG Nonlinear Conjugate Gradient.
NN Neural Network.

Octet Octet Truss.

PANN Physics-Augmented Neural Network.
PBCs Periodic Boundary Conditions.
PDE Partial Differential Equation.
PINN Physics-Informed Neural Network.
POD Proper Orthogonal Decomposition.

pROM	projection-based Reduced Order Model.
PSO	Particle Swarm Optimization.
RB	Reduced Basis.
RBCC	Reinforced Body-Centered Cubic.
RBF	Radial Basis Function.
RMSE	Root Mean Squared Error.
RNN	Recurrent Neural Network.
ROM	Reduced Order Model.
RSM	Response Surface Method.
RVE	Representative Volume Element.
SA	Simulated Annealing.
SC	Simple Cubic.
SEA	Specific Energy Absorption.
SLA	Stereolithography.
SLM	Selective Laser Melting.
TCO	Truncated-cuboctahedron.
TO	Topology Optimization.
TPMS	Triply periodic Minimal Surface.

Chapter 1

Introduction

"On voyage pour changer, non de lieu, mais d'idées." — Hippolyte Taine

1.1 Motivations and Objectives

Humans have always sought to develop new materials and use them to create tools that advance their capabilities. In this quest, the requirement for processes enabling the fabrication of complex geometries has significantly influenced the development of modern manufacturing techniques. Additive manufacturing is a technology that has revolutionized the possibilities of design. It has accelerated the development of novel structures with enhanced or unprecedented properties. Nevertheless, it has also introduced new challenges in the design process due to the increased structural complexity. The design space is so vast that new structural concepts, called lattice structures, have emerged to simplify the design process by introducing repetitive patterns of unit cells. This approach allows for the creation of complex geometries while maintaining a certain level of control over the mechanical properties of the structure.

The benefit of lattice structures is twofold: they facilitate significant weight reduction and provide a framework for tailoring multifunctional and multiphysical properties. Their capabilities are particularly advantageous in a context of sustainability, where reducing material usage and fuel consumption is critical, especially in transportation and mobility sectors. They are also well known for their ability to absorb energy, which is particularly valuable in applications such as protective equipment (see the recently developed application in Figure 1.1).

However, this design freedom comes at the expense of increased computational cost from a numerical point of view when dealing with such complex structures. Each step, from geometric modeling to the creation of the printing file, remains computationally



Fig. 1.1 Innovative use cases of lattice structures in (a) Riddell American football helmet (*Riddell Diamond Technology* 2019), (b) Japan Aerospace Exploration Agency (JAXA) moon lander (*JAXA Moon Lander* 2024), (c) Adidas running shoe (*Adidas 4DFWD* 2021).

intensive unless dedicated software or hardware adaptations are introduced. New methodologies are therefore required to reduce the computational burden associated with both the simulation and the optimization of these systems. This will enable the exploration of a wider range of design options within a shorter time frame, ultimately leading to more efficient and effective designs.

To address these challenges, our work focuses on developing innovative design methods, which plays a key role in developing structures with desired properties. The problem is multifaceted and intertwined, as it involves the concurrent consideration of the geometry type, the simulation method, and the optimization algorithm. Each of these components can significantly influence the others, leading to a complex interplay that must be carefully managed to achieve efficient and effective design solutions. Our research has led us to first develop a method for simulating the mechanical behavior of lattice structures using beam elements with specific joint modifications. Subsequently, building on this simulation method, we have established a versatile optimization framework suitable for addressing diverse lattice architectures, parameterizations, and structural

design objectives. The framework is designed to be sufficiently flexible to accommodate multiphysics analyses, multiscale modeling, and multiple lattice cell types, in order to tackle a wide range of design challenges. We expect that our method will facilitate the design of multifunctional lattice structures in application domains that offer positive societal benefits. The contributions of this work to the research field are distributed throughout the document and are summarized in the list of publications and scientific presentations provided in Section 1.3.

1.2 Outline

To guide the reader through the contributions of this work, the document is structured as follows (See Figure 1.2). Chapter 2 presents an in-depth review of the current state of the art concerning lattice structures, including aspects related to their design, physical behavior, simulation methodologies, and optimization approaches. Particular attention is given to surrogate modeling techniques, which are instrumental in reducing computational costs during the optimization phase. Chapter 3 focuses on the simulation aspects, introducing a joint stiffening strategy for beam finite elements. This approach aims to improve the accuracy of mechanical predictions in lattice structures, which are frequently underestimated by conventional beam formulations due to weak joint representation. Relying on these simulation capabilities, Chapter 4 presents an enhanced strategy that couples domain decomposition with surrogate modeling techniques, to further improve computational efficiency and scalability. Decomposing large-scale problems into smaller and more manageable subproblems is a well-established principle in numerical analysis. However, this approach has not yet been widely applied in the context of lattice structures, even though their multiscale nature makes them particularly well suited for such a strategy. By embedding surrogate models, also known as metamodels or response surface models, within this framework, the computational efficiency is further enhanced while ensuring compatibility with gradient-based optimization processes. The benefits of the proposed simulation methodology are demonstrated in Chapter 5, which introduces an optimization framework in which independent geometries can be assigned to each cell. This integration of domain decomposition and surrogate modeling provides a powerful tool for the efficient optimization of complex lattice structures, leveraging the complementary strengths of both approaches. Finally, Chapter 6 concludes the thesis by summarizing the main contributions of this work and discussing future research directions.

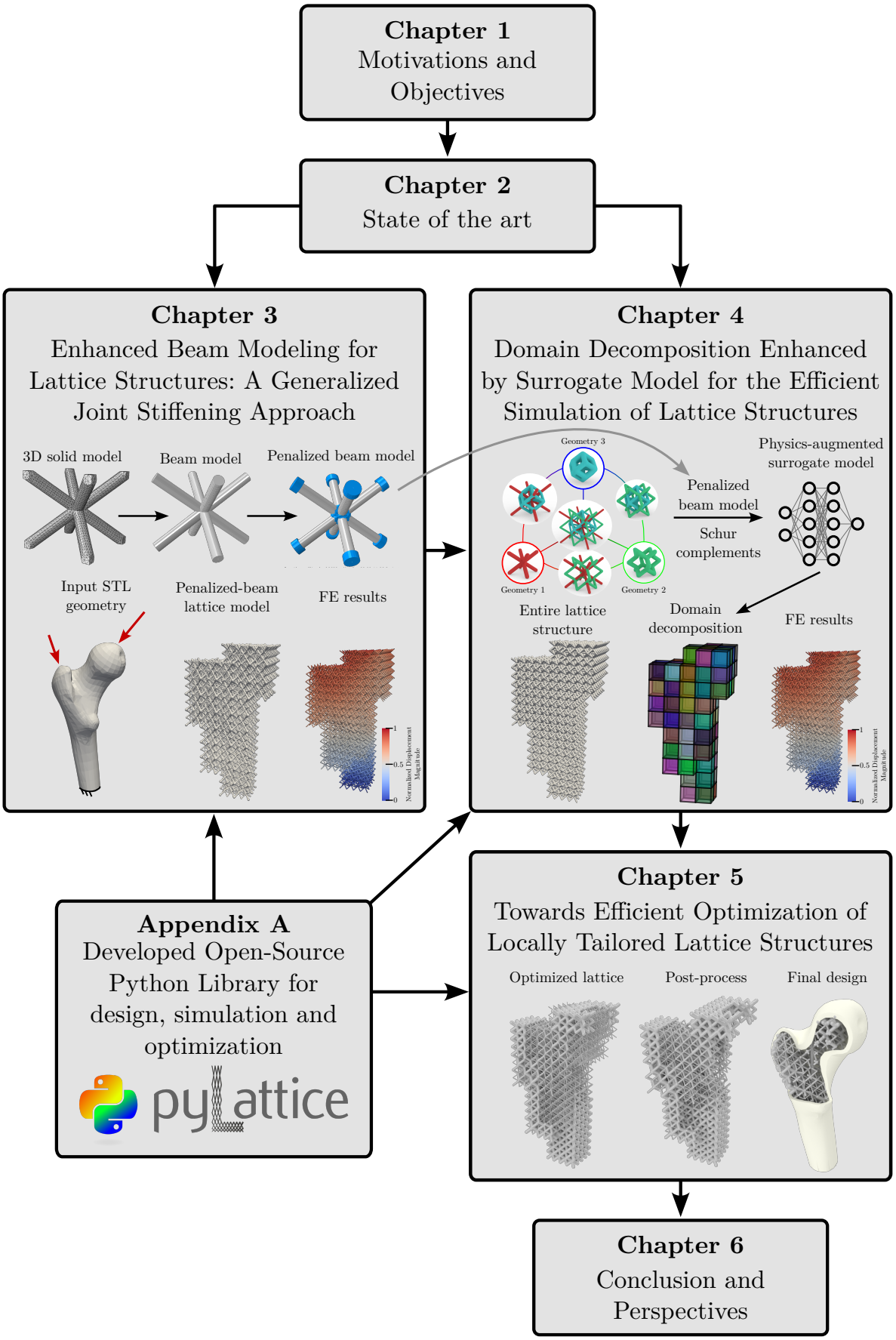

Fig. 1.2 Schematic representation of the thesis chapter organization.

1.3 Contributions

This thesis led to the following publications and presentations.

Peer-review journals

Cadart, T., T. Hirschler, S. Bahi, S. Roth, F. Demoly, and N. Lebaal (2025). "An Optimal Penalty Method for the Joint Stiffening in Beam Models of Additively Manufactured Lattice Structures". In: *International Journal of Solids and Structures* 306, p. 113107.

Peer-reviewed journals in preparation

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (n.d.[a]). "A PANN-enhanced Domain Decomposition Method for the Optimal Design of Additively Manufactured Lattice Structures". In: *preparation* ().

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (n.d.[b]). "pyLattice: An Open-Source Python Package for Design and Simulation of Lattice Structures". In: *preparation* ().

Conference papers and presentations

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025a). "Conception numérique de structures lattice par décomposition de domaine et réseau neuronal". In: *19ème colloque national S.mart*. Vogüé, Ardèche, France.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025b). "Optimal Design of Lattice Structures via PINN-based Domain Decomposition". In: *The Fifth International Conference on Simulation for Additive Manufacturing*. Pavia, Italy.

Cadart, T., T. Hirschler, S. Bahi, S. Roth, F. Demoly, and N. Lebaal (2024). "Une modélisation poutre enrichie des structures lattices en fabrication additive". In: *16e colloque national en calcul des structures*. Giens, Var, France.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025c). "A PINN-based Domain Decomposition Method for the Optimal Design of Additively Manufactured Lattice Structures". In: *12th European Solid Mechanics Conference*. Lyon, France.

Chapter 2

A Global Vision for Optimal Design of Lattice Structures

"The smallest leaf outside our window reveals how weak and childlike human science truly is."

— Attributed to Albert Einstein

Contents

2.1 Lattice Structures Design Driving the Overall Performance . . . 8
2.1.1 A Large Variety of Lattice Structures . . . 9
2.1.2 Multiphysical Properties of Lattice Structures . . . 12
2.1.3 Manufacturing Strategies and Constraints for Lattice Structures . . 15
2.1.4 Opportunities and Challenges in Generative Design . . . 17
2.1.5 Multidisciplinary Applications of Lattice Structures . . . 19
2.2 Efficient Property Evaluation for Lattice Structures . . . 21
2.2.1 Single-Scale Modeling Approaches . . . 21
2.2.2 Multiscale and Decomposition Strategies . . . 26
2.2.3 Experimental Characterization . . . 31
2.3 Optimal Design for Mechanical Lattice Structures . . . 34
2.3.1 Optimization Strategies at the Cell-Scale . . . 35
2.3.2 Optimization Strategies at the Macro-Scale . . . 37
2.3.3 Optimization Strategies across Multiple Scales . . . 39
2.3.4 Metaheuristic Strategies . . . 40
2.4 Accelerating Optimization with Surrogate Models . . . 43
2.4.1 Classification of Surrogate Modeling Approaches . . . 43
2.4.2 Levels of Surrogate Model Integration . . . 53
2.4.3 Potential and Limitations of Surrogate Modeling . . . 58
2.5 Scientific Challenges and Positioning . . . 59

2.1 Lattice Structures Design Driving the Overall Performance

Historically, the field of materials science has focused on enhancing material performance by tuning micro-structural features, thereby tailoring mechanical, thermal, or electrical properties to suit specific application requirements. While significant progress has been achieved in the engineering of metallic, polymeric, and ceramic materials, these improvements have remained limited to the microscale. This limitation arises because traditional fabrication strategies rely on uniform material distributions and have limited ability to exploit spatially varying architectures to modulate global performance. In addition, conventional manufacturing processes such as casting, machining, or injection molding have restricted achievable geometries due to the need for molds, tooling, and standardized workflows. These limitations not only reduced design freedom, but also required substantial experimental effort to validate component performance. To overcome these constraints and expand the design possibilities, researchers have turned their attention to cellular structures that allow for a decoupling between material composition and structural response.

Foams as cellular solids were initially developed. These structures are stochastic materials composed of a random distribution of pores or cells within a solid matrix (Frisch 1981). Using either foaming or sintering processes, these materials offer low density, high energy absorption, and good damping properties. However, their structural randomness limits the degree of control over local properties and makes predictive modeling challenging (Hössinger-Kalteis et al. 2021; Köll and Hallström 2016).

Recent advances in Additive Manufacturing (AM) have dramatically expanded the design space (I. Gibson et al. 2015). By offering unprecedented design freedom in terms of geometry, hierarchical structuring, and material distribution, AM allows the fabrication of architected structures that are beyond the capabilities of traditional manufacturing methods. This technological shift enables simultaneous design on both the macroscopic scale of the structural component and the microscopic scale of the bulk material, therefore offering unprecedented control over the resulting properties. However, it significantly increases the complexity of the design process. In the following, we provide an overview of the wide variety of lattice structures, their associated multiphysical properties, the manufacturing techniques used to produce them, and the related design challenges. The section concludes with application cases that demonstrate the transformative potential of lattice structures in diverse engineering fields.

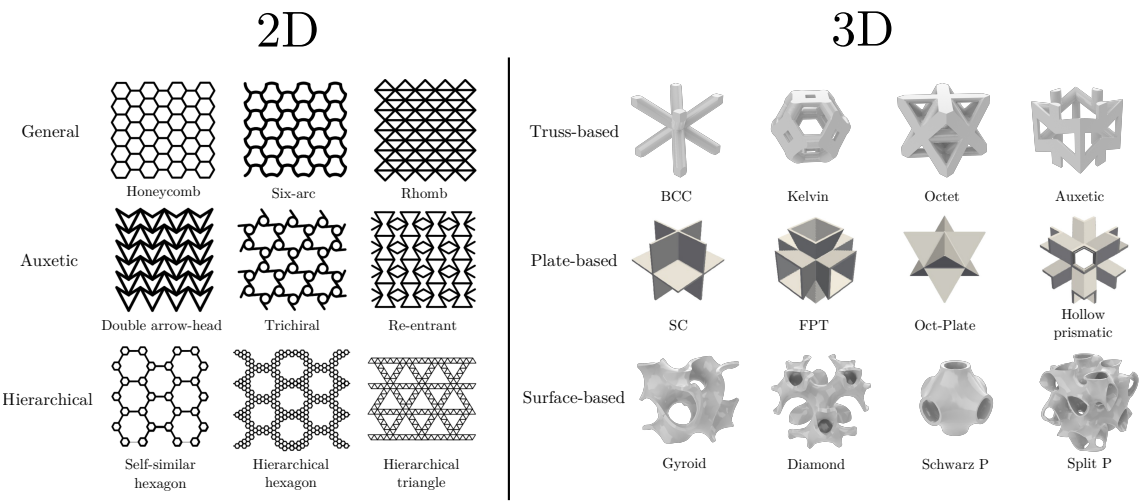

Fig. 2.1 Lattice types and related examples. (2D lattice adapted with permission from (Yin, W. Zhang, et al. 2023), Plate-based adapted with permission from (B. Sun et al. 2023)).

2.1.1 A Large Variety of Lattice Structures

Lattice structures, often seen as the deterministic counterpart of foams, represent a class of architected materials composed of periodic or aperiodic spatial arrangements of solid and void regions (Seharing et al. 2020; Y. Wu, Fang, et al. 2023). Their geometry can be precisely engineered to tailor a wide range of effective properties, including stiffness, strength, energy absorption, thermal conductivity, and electrical permittivity, thereby enabling multifunctional performance. Lattice structures can be broadly classified according to their dimensionality and topological complexity into 2D, and 3D architectures, each offering different levels of design flexibility and mechanical performance (see Figure 2.1).

2D lattices are planar arrangements of struts or walls, typically used in sandwich panels or surface reinforcement layers (R. Adams et al. 2022; Y. Wang et al. 2020; Zhou et al. 2025). They are easy to design and fabricate, but their out-of-plane mechanical behavior is often limited.

3D lattices, on the other hand, exhibit full spatial connectivity and can be designed to exhibit complex behaviors such as negative Poisson's ratio, tailored stiffness anisotropy, or programmable deformation patterns (Yin, W. Zhang, et al. 2023). They include truss-based, plate-based, and surface-based architectures such as Triply periodic Minimal Surface (TPMS).

The important feature of this class of materials is that their effective behavior is governed not only by the constituent material but also by the topology of the unit cell. This dual

control over performance, through both material selection and structural design, makes it possible to engineer novel properties, such as ultralight stiffness or enhanced energy absorption, that are inaccessible to bulk materials alone.

Bio-inspiration In the exploration of even more efficient designs, nature itself offers an open and abundant library of structural solutions. Indeed, bio-inspiration has become a powerful paradigm in the development of advanced architected materials (Ha and Lu 2020; Siddique et al. 2022). Natural systems, such as bones, corals, or plant cellular frameworks, exhibit remarkable mechanical and multifunctional performances that are the result of millions of years of evolutionary optimization. By mimicking these biological architectures, researchers aim to discover new structural behaviors and improve performance criteria such as strength-to-weight ratio, and damage tolerance, to name a few.

Graded lattice structures The design possibilities at the cell level are extensive, but additional complexity can also be introduced at the structure level. A promising strategy to enhance the mechanical performance of lattice structures is the application of spatial gradients (Y. Li et al. 2020). Gradients can be introduced in various ways including:

- Cell relative density (X. Liu et al. 2022; Rodrigo et al. 2023; J. Wang, J. Zhu, et al. 2023; J. Wang, Westermann, et al. 2024; C. Yang et al. 2021)
- Cell size (J. Yang et al. 2022)
- Cell material properties (Montgomery et al. 2023; Rahman et al. 2021; Valizadeh and Weeger 2024)
- Cell geometry type (Ozdemir et al. 2023; Vafaeefar et al. 2025)
- Or a combination of all the above (Rahman et al. 2021)

Beyond geometric gradients, spatial variation in material composition offers another powerful strategy to tailor the mechanical response of lattice structures. By locally tailoring material properties within the architecture, one can achieve enhanced or heterogeneous behaviors that would be difficult to achieve with uniform materials. AM techniques such as Fused Filament Fabrication (FFF) or material jetting allow the use of multiple materials within a single part. For example, by combining soft and hard polymers in honeycomb architectures, Rahman et al. (2021) demonstrated enhanced energy absorption and structural stiffness. Vat Photopolymerization techniques, such as Stereolithography (SLA) or Digital Light Processing (DLP), offer further control by

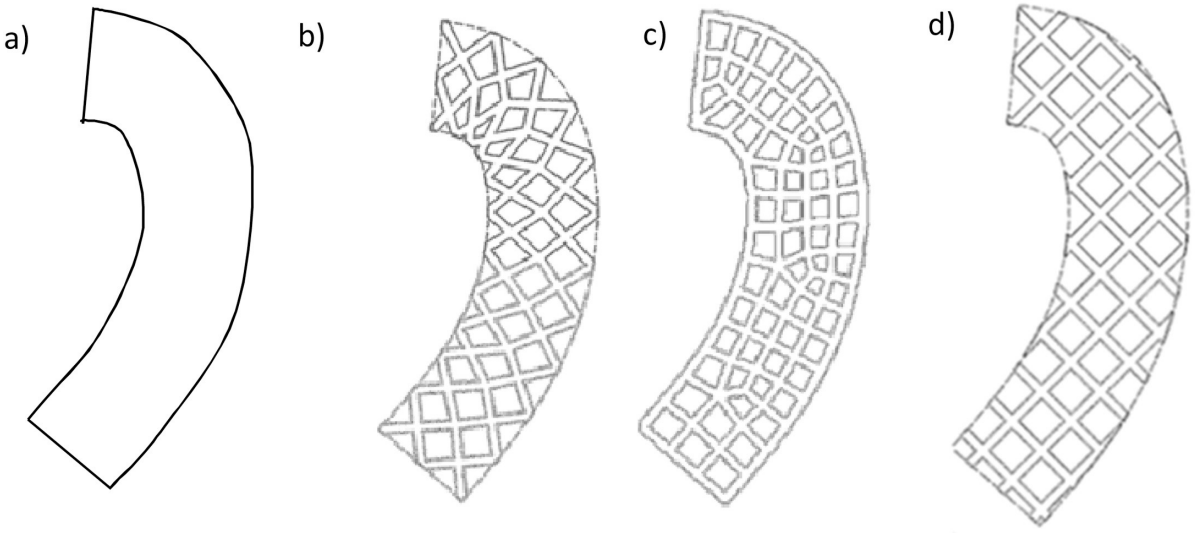

Fig. 2.2 Lattice generation methods: a) domain boundary, b) swept, c) meshed, and d) trimmed. Adapted with permission from (Aremu et al. 2016).

enabling local tuning of the polymerization degree. This capability can be strategically used to locally adjust material properties and enhance the overall structural behavior (Montgomery et al. 2023; Valizadeh and Weeger 2024). However, the introduction of gradients must be carefully managed to avoid discontinuities. Significant variation between neighboring cells can compromise their connectivity. This issue can be mitigated by defining a global function over the entire structure to control the gradient and evaluate it on each element composing the lattice, as proposed for truss-based lattices by Nian et al. (2021). The challenge becomes more complex for TPMS structures, where both the relative density and the surface continuity must be preserved. To address this, Ozdemir et al. (2023) proposed a generalized blending algorithm capable of ensuring continuity between different TPMS cell types.

Lattices in complex geometries Due to their simplicity of design and ease of manufacturing, quadrilateral-based lattices are used in most of studies. While they are convenient in academic research, integrating these lattices within complex parts remains challenging in practice. Multiple methods have been proposed to generate lattice structures within complex geometries (H. Zhong, Mo, et al. 2024), as illustrated in Figure 2.2. The most straightforward approach consists in trimming a regular lattice to the design domain of the part (Aremu et al. 2016). Although this method is straightforward to implement, it may significantly degrade mechanical performance due to the presence of discontinuities at the trimmed boundaries. A second category of methods relies on

meshing strategies, which can be subdivided into two main variants. The first variant consists in reinterpreting the existing mesh elements as beam elements by assigning them a prescribed thickness (Terriault and Brailovski 2018; Vaiani et al. 2025). The second variant embeds a predefined lattice geometry within the volumetric mesh elements (F. Liu, M. Chen, et al. 2024). A third and more sophisticated method is the swept approach (Dal Fabbro et al. 2021; Hirschler, Antolin, et al. 2022; Hirschler, Bouclier, et al. 2024). Although this method is more complex to implement, it enables greater control over the lattice geometry. The underlying idea is to "sweep" the lattice pattern along a guiding path or surface, ensuring alignment with the curvature and features of the domain. However, incorporating lattice geometry in complex parts is not without consequences from a mechanical performance point of view. D. Li, Liao, et al. (2020) compared the mechanical performance of conformal gradient lattices to those of classical quadrilateral lattices. They demonstrated that conformal lattices offer superior mechanical properties, primarily because the lattice architecture better aligns with the stress field within the structure.

2.1.2 Multiphysical Properties of Lattice Structures

The properties of lattice structures are influenced by several factors, including:

- The properties of the constituent material.
- The geometry of the unit cell.
- The relative density $\rho_{\text{rel}} = V_{\text{material}}/V_{\text{cell}}$, where V_{material} is the volume of solid material and V_{cell} is the total volume of the unit cell.

Among these, the geometry of the unit cell plays a central role in determining the overall behavior of the lattice. In the case of mechanical properties, lattice structures can be broadly categorized into two behavioral types, namely, bending-dominated and stretching-dominated. Bending-dominated lattices are characterized by low connectivity, where the struts are primarily subjected to bending moments. In contrast, stretching-dominated lattices have high connectivity, and the struts are mainly loaded under axial tension or compression (M. Ashby 2006). For pin-jointed beam networks, the type of mechanical behavior can be predicted using Maxwell's stability criterion (Maxwell 1864). The condition for a structure to be both statically and kinematically determinate is given by:

$$M = b - 2j + 3 = 0 \quad \text{(in 2D)} \tag{2.1}$$

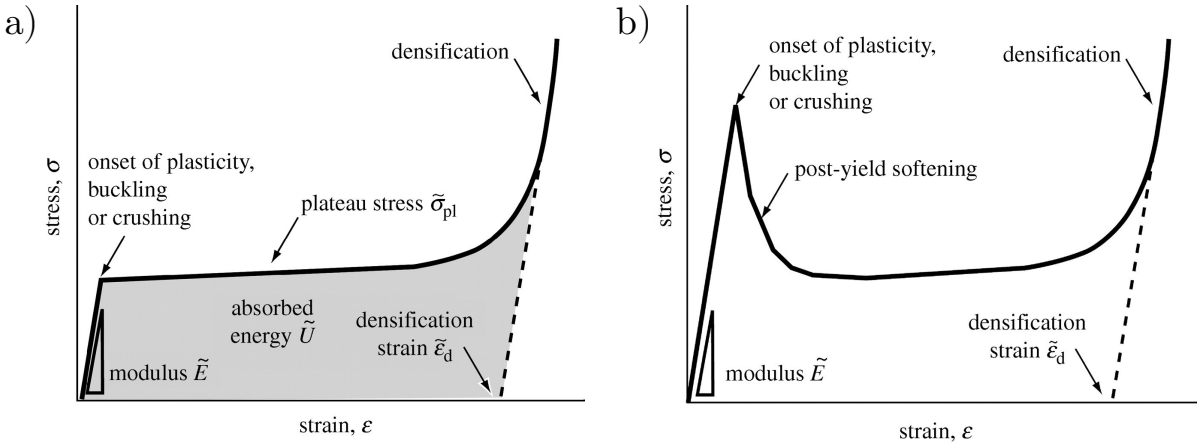

Fig. 2.3 Schematic stress–strain curves for (a) bending-dominated and (b) stretching-dominated structures. Adapted with permission from (M. Ashby 2006).

$$M = b - 3j + 6 = 0 \quad \text{(in 3D)} \tag{2.2}$$

where b is the number of struts, j the number of frictionless joints, and M the number of internal mechanisms. If $M > 0$, the structure is over-constrained and is considered stretching-dominated. If $M < 0$, the structure is under-constrained and is considered bending-dominated.

These two classes of structures exhibit markedly different stress–strain behaviors, as illustrated in Figure 2.3. Bending-dominated lattices, illustrated in Figure 2.3(a), typically show an initial linear elastic region followed by a long stress plateau corresponding to progressive plastic collapse or elastic buckling of struts, until densification occurs and the stress rises sharply. On the other hand, in Figure 2.3(b), stretching-dominated lattices present a stiffer linear elastic phase. Due to the axial loading of struts, failure generally occurs via plastic buckling or brittle fracture, often leading to post-yield softening.

For lightweight structural applications, stretching-dominated lattices are usually preferred because they provide a higher specific stiffness and strength during the elastic regime. Among the various design parameters, the relative density is the most widely studied in relation to mechanical properties. While early studies assumed a linear relationship between relative density and effective properties, it is now well established that this relationship is generally non-linear (Almesmari et al. 2023; M. Ashby 2006; Jalali et al. 2024).

Among the various interesting properties of lattice structures, one yields a high interest, the energy absorption (Y. Wu, Fang, et al. 2023; Yin, W. Zhang, et al. 2023). Energy absorption refers to the capacity of a material or structure to dissipate mechanical energy under loading, typically through irreversible deformation mechanisms such as

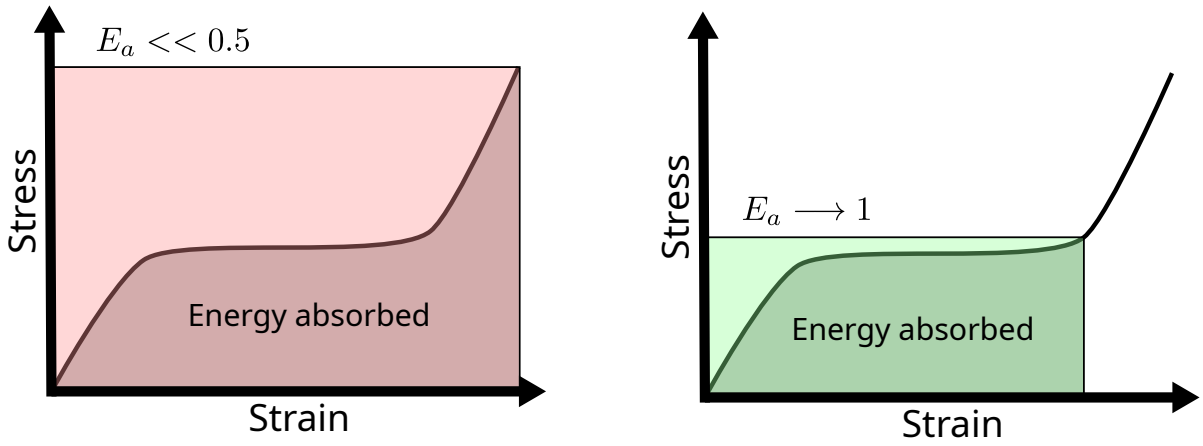

Fig. 2.4 Schematic of the energy absorption definition.

plasticity, buckling, or fracture. In the context of lattice structures, energy absorption is evaluated during compressive loading, where the structure deforms progressively while absorbing the input energy. The absorbed energy E_{a} is calculated as the area under the force–displacement curve, or equivalently the stress-strain curve, up to a given deformation level:

$$E_{\mathrm{a}} = \int_0^{\varepsilon_{\max}} \sigma(\varepsilon)\, d\varepsilon \tag{2.3}$$

where $\sigma(\varepsilon)$ is the stress as a function of strain, and $\varepsilon_{\max}$ denotes the maximum strain. However, this classical definition does not account for the peak stress transmitted through the structure, which may be critical in applications where excessive force could damage the protected object (Duncan et al. 2018) (See Figure 2.4). A more relevant criterion consists in normalizing the absorbed energy by the product of the maximum stress $\sigma_{\max}$ and strain $\varepsilon_{\max}$:

$$E_{\mathrm{a}} = \frac{\int_0^{\varepsilon_{\max}} \sigma(\varepsilon)\, d\varepsilon}{\sigma_{\max}\varepsilon_{\max}} \tag{2.4}$$

To enable comparison between different structures or materials, the Specific Energy Absorption (SEA) is often introduced, defined as the energy absorbed per unit mass:

$$SEA = \frac{E_{\mathrm{a}}}{m} \tag{2.5}$$

where m is the mass of the structure. This metric is particularly important for lightweight applications, where maximizing energy absorption while minimizing structural mass is crucial. With these definitions, bending-dominated lattices are generally more favorable for energy absorption purposes due to their long stress plateau under compression, enabling gradual energy dissipation without significant stress increase.

Beyond mechanical performance, lattice structures also exhibit advantageous thermal (Sajjad et al. 2022), vibrational (Al Rifaie et al. 2022), and acoustic (Al Rifaie et al. 2022) properties, which broaden their applicability in multifunctional and multiphysics design. Thermally, the effective conductivity of a lattice can be tailored by adjusting the relative density and topology of the unit cell, enabling the design of heat exchangers or thermal insulators with optimized mass efficiency (Ai and X. Gao 2018). Vibrational properties, such as natural frequencies and mode shapes, are strongly influenced by the geometry and connectivity of the lattice network. This makes them suitable for applications requiring vibration damping or bandgap engineering, such as in phononic crystals or vibration isolators (Scalzo et al. 2021). Acoustically, lattices can be designed to attenuate or manipulate sound propagation by exploiting mechanisms such as local resonances, or porosity-induced dissipation. Recent studies have demonstrated the potential of architected lattices to serve as lightweight sound absorbers or acoustic metamaterials with tunable frequency responses (Chua et al. 2023).

2.1.3 Manufacturing Strategies and Constraints for Lattice Structures

Lattice structures are inherently complex, multiscale architectures, which makes their fabrication particularly challenging. Often, the manufacturing process itself becomes a limiting factor in their design. Traditional manufacturing methods have been employed to produce lattice structures. Two-dimensional lattices are easier to manufacture; for example, X. Xu et al. (2022) used wire-cut electrical discharge machining to manufacture augmented honeycombs inspired from beetle elytra. An interesting subclass of 2D lattices that can be manufactured with planar processes are origami-based lattices (Harris and McShane 2021; H. Liu, Qiao, et al. 2023; Tomita et al. 2023; J. Xu et al. 2021). Van Manen et al. (2023) has developed a method to manufacture origami-based lattices automatically from flat titanium sheets. They use a laser cutting process to cut the sheets and then fold them into the desired shape. For three-dimensional lattices, Druschitz et al. (2016) employed metal casting to manufacture octet truss geometries, demonstrating the feasibility of using formative processes for architected materials.

However, the most widespread and versatile method for fabricating complex and customized lattice structures is a class of technologies known as Additive Manufacturing (AM). AM refers to a family of processes that build parts layer by layer from digital models, offering unprecedented design freedom compared to traditional subtractive or formative techniques. Initially developed in the 1980s for rapid prototyping (André

et al. 1984), AM has since matured into a suite of industrial manufacturing technologies applicable to polymers, metals, ceramics, and composites. Its layer-wise approach is particularly well suited for fabricating architected materials, enabling the realization of intricate lattice structures with controlled topology, relative density, and material composition. Among the various AM techniques, several have emerged as particularly well suited for the fabrication of lattice structures, each offering specific advantages and limitations depending on the target material, resolution requirements, and geometric complexity.

Fused Filament Fabrication (FFF) is an extrusion-based AM technique in which thermoplastic filaments are melted and deposited layer by layer through a heated nozzle (Peng et al. 2022). It is widely used for polymer lattices due to its low cost, ease of operation, and scalability, making it suitable for prototyping and the fabrication of large parts. However, FFF offers limited printing resolution and often exhibits anisotropic mechanical properties arising from poor interlayer adhesion and path-dependent deposition, which may significantly compromise the integrity of fine lattice features (Dong et al. 2018).

Direct Ink Writing (DIW) involves the extrusion of viscoelastic inks through a nozzle to create complex geometries layer by layer, often at room temperature (Saadi et al. 2022). It enables the fabrication of lattice structures using soft, conductive, or responsive materials, and supports multi-material integration with minimal thermal stress. On the downside, Direct Ink Writing (DIW) processes are typically slow and highly sensitive to ink rheology, requiring extensive formulation to achieve printability and structural stability during the build.

Stereolithography (SLA) and **Digital Light Processing (DLP)** are vat photopolymerization-based processes in which liquid resin is selectively cured by a light source to build the part (Pagac et al. 2021). These techniques provide excellent resolution and surface finish, making them particularly suitable for manufacturing complex polymer lattices. Their main limitations lie in their limited material choices, as they are confined to photopolymers that may exhibit lower mechanical strength, thermal stability, and long-term durability compared to thermoplastics or metals.

Selective Laser Melting (SLM) and **Electron Beam Melting (EBM)** are powder bed fusion techniques in which a high-energy beam selectively melts metallic powder to form solid layers (Riva et al. 2021). These methods enable the fabrication of complex metallic lattices with excellent mechanical strength and are widely employed in demanding sectors such as aerospace and biomedical engineering. Nevertheless, they are affected by high residual stresses, surface roughness, and process-induced defects such as

porosity or lack of fusion (Echeta, Feng, et al. 2020), which can negatively impact fatigue life and dimensional precision. Moreover, the removal of unmelted powder from internal cavities remains a major challenge, particularly for small or enclosed lattice features.

Material Jetting is a droplet-based process in which build materials are selectively deposited and cured using UV light to form solid parts (Mora et al. 2022). This technique allows multi-material fabrication with high resolution and smooth surface finish, enabling functional grading and localized property control. However, it is currently limited by the narrow range of printable materials, the relatively low mechanical properties of the resulting parts, and the impossibility of printing mid-air, which often requires extensive support structures (Pugalendhi et al. 2021). This significantly constrains its applicability in industry.

Defects in AM A common issue across all AM methods is the presence of defects, which can significantly impact the mechanical, thermal, and electrical properties of printed parts (Bai et al. 2024; Niutta et al. 2022; Prajapati et al. 2023). Powder bed fusion techniques are particularly susceptible to defects such as porosity, lack of fusion, and residual stresses (Beevers et al. 2025; Echeta, Feng, et al. 2020). To address these challenges, various post-processing techniques (Safari et al. 2022) and optimizations of printing parameters (Kothandaraman and Balasubramanian 2024) have been proposed. Moreover, several studies have investigated the influence of such defects (Echeta, Dutton, et al. 2021; D. Li, Qin, B. Chen, et al. 2021) and developed uncertainty quantification methods (Korshunova, G. Alaimo, et al. 2021; Korshunova, Papaioannou, et al. 2021) to improve the predictive accuracy of lattice models.

2.1.4 Generative Design: Opportunities for Complex Geometries and Related Challenges

Designing complex lattice structures remains a major challenge for conventional Computer-Aided Design (CAD) software such as SolidWorks or CATIA, primarily due to the multiscale nature of these architectures and the resulting computational burden. To exploit the inherent quasi-periodicity of unit cells, new design methodologies have emerged, aiming to streamline the generation of highly complex geometries.

Generative design refers to a design paradigm where algorithms are used to autonomously generate geometries based on predefined parameters, constraints, and functions (Jaisawal and Agrawal 2021). It enables the efficient exploration of vast design spaces by varying a relatively small set of parameters, facilitating the creation of in-

tricate structures that would be impractical to model manually. To further tackle the challenges associated with representing highly detailed and smooth geometries, implicit modeling has gained increasing attention. Unlike conventional boundary representation (B-Rep) techniques, implicit modeling describes geometries as level sets of scalar fields, offering several advantages, such as inherent geometric continuity, smooth transitions, and the ability to represent extremely complex topologies without the need for explicit meshing. This approach is particularly well suited for lattice structures with conformal, graded, or continuously varying properties, where traditional representations would lead to prohibitive computational costs. Several software solutions now integrate generative design and implicit modeling capabilities, including Rhinoceros 3D combined with its Grasshopper plugin (Robert McNeel 2025), Fusion 360 (Autodesk 2025), Creo (PTC 2025), nTop (formerly nTopology) (nTop 2025), or Carbon Design Engine (Carbon 2025).

Despite these advances, bridging the gap between generative design and simulation remains challenging. Classical simulation approaches require the generation of a conformal mesh that precisely matches the part geometry, a process that becomes cumbersome for highly complex lattice structures. To overcome these limitations, new simulation strategies have emerged, often referred to as mesh-independent approaches. These methods allow simulations to operate directly on implicit geometries without explicit meshing, enabling faster and more robust analyses of intricate designs. Details on these methods will be discussed further in Section 2.2.1.

Beyond numerical simulation, the transition from generative design to AM presents similar challenges, particularly regarding the complexity and size of the generated meshes. Traditional AM workflows rely on the generation of a surface mesh, typically in the form of an STL file format, which explicitly discretizes the geometry boundaries. This surface mesh is then processed using slicer software, such as Ultimaker Cura, to generate the toolpath required by the printer. For highly complex lattice structures, this process results in extremely large mesh files, often comprising millions of facets, leading to increased memory requirements, long processing times, and potential loss of geometric fidelity due to discretization artifacts. Moreover, the large size of the files can create bottlenecks during slicing and printing preparation, and may exceed the capabilities of standard 3D printers or slicer software. Some studies have proposed methods to reduce the size of STL files by optimizing the mesh construction process (Chougrani et al. 2017). An alternative approach consists in bypassing explicit mesh generation altogether by directly converting implicit geometries into printable formats (S. Liu et al. 2021). In these approaches, the geometry is defined as a continuous scalar field, and the slicing process operates directly on the implicit representation. This strategy not only reduces

the size of the generated files but also preserves the smoothness and accuracy of the original design model, enabling the fabrication of highly detailed structures with reduced computational overhead. Several software tools now offer implicit-to-print workflows, including:

- nTop (nTop 2025), which can export implicit geometries directly in a slice-based format compatible with various AM technologies,
- Carbon's software platform (Carbon 2025), which uses voxel-based representations derived from implicit models to control the printing process with high precision.

2.1.5 Multidisciplinary Applications of Lattice Structures

Innovative properties and functionalities, combined with recent advances in design and manufacturing techniques, have led to a wide range of applications for lattice structures (Arredondo-Soto et al. 2021; Khan and Riccio 2024; Pan et al. 2020; Sajjad et al. 2022). Numerous companies have emerged in recent years, developing new products that leverage the capabilities of lattice architectures, as illustrated by several examples in Figure 2.5. A pioneer enterprise in this field is Adidas, which integrated a 3D-printed lattice midsole into its running shoes (*Adidas 4DFWD* 2021), offering enhanced cushioning and energy return. In the helmet industry, Riddell (*Riddell Diamond Technology* 2019) developed an internal 3D-printed lattice that improves impact absorption and energy dissipation, significantly enhancing player safety. In cycling, brands such as Specialized (*Specialized and Carbon Saddle* 2020) and Fizik (*Fizik 3D* 2022) have introduced 3D-printed saddles that enhance comfort and performance by optimizing pressure distribution in the perineal area. Companies, such as Carbon, are at the forefront of 3D-printed lattice technology, offering solutions to help businesses design and rapidly manufacture lattice-enhanced products. Lattice structures are also highly promising in biomedical applications, particularly for bone implants, where they can mimic the mechanical properties of natural bone while promoting osseointegration and tissue regeneration (*Lattice Medical* 2023). In the aerospace sector, lattice structures are investigated to reduce component weight and increase fuel efficiency in aircraft and spacecraft applications.

Conclusion on New Opportunities for Lattice Structures

Lattice structures, and more generally complex geometries, have attracted considerable attention, leading to the development of many solutions for their efficient modeling, simulation, and manufacturing. These solutions will serve as a foundation for the

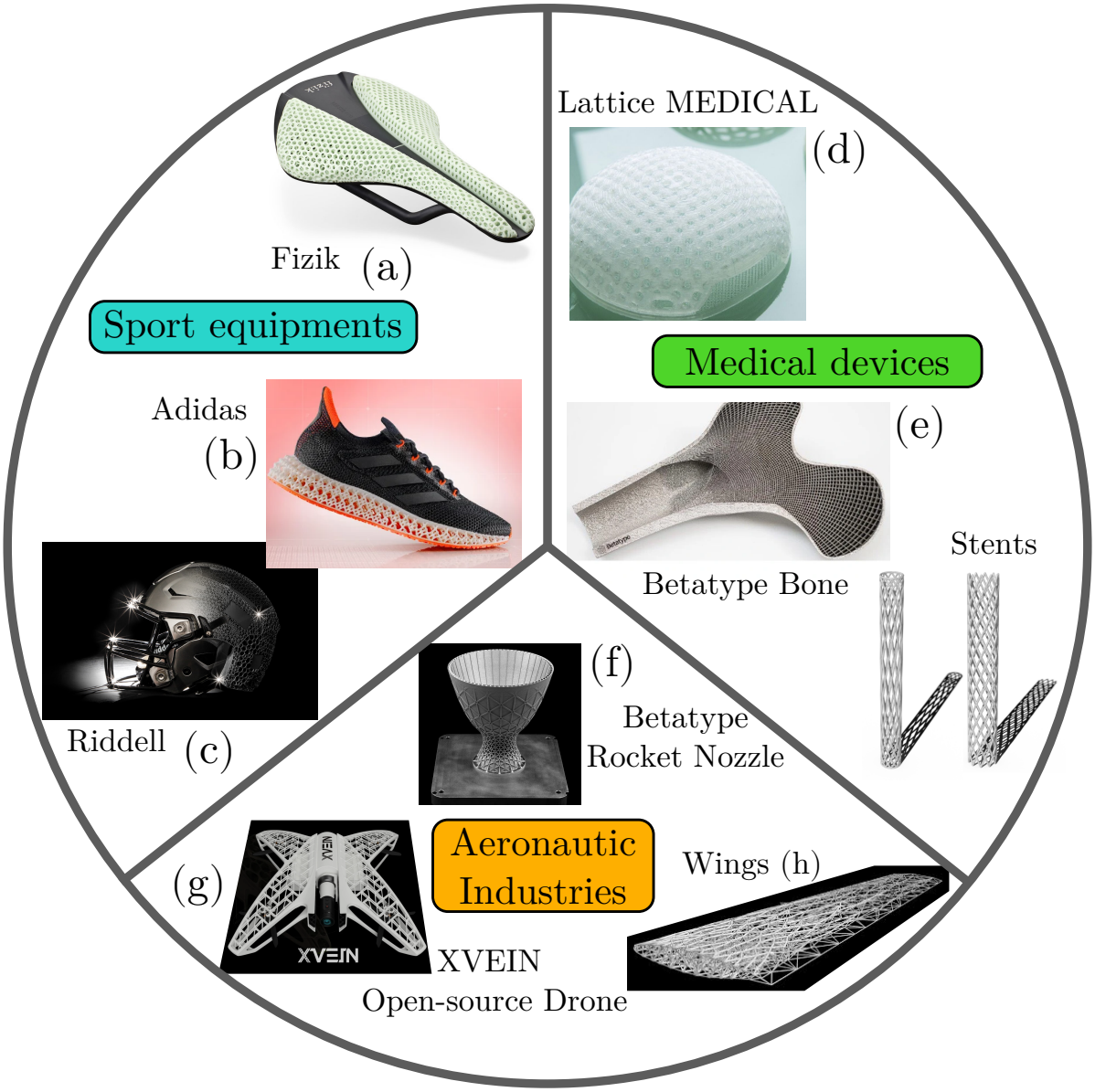

Fig. 2.5 Overview of applications of lattice structures in various fields. Adapted with permission from (a) (*Fizik 3D* 2022), (b) (*Adidas 4DFWD* 2021), (c) (*Riddell Diamond Technology* 2019), (d) (*Lattice Medical* 2023), (e) (*Orthopedic Implants AM* 2024), (f) (*Space Industry AM* 2024), (g) (*X VEIN* 2020), (h) (Opgenoord and Willcox 2018).

development of new tools in the coming years, enabling the creation of increasingly specialized structures tailored to specific, multiphysical, and multiscale applications. The design freedom offered by new manufacturing processes opens the door to novel possibilities. For instance, structures can be adapted to user-specific needs, such as customized orthopedic soles conforming exactly to foot morphology or personalized helmets optimized for individual head shapes.

2.2 Efficient Property Evaluation for Lattice Structures

The accurate and efficient evaluation of material and structural properties is essential for identifying opportunities to improve the performance of lattice structures. Various approaches are available, ranging from analytical models and experimental characterization to advanced numerical simulations. Each method offers distinct advantages and limitations, and the choice of the approach depends on factors such as the lattice geometry type, the arrangement of unit cells, and the targeted performance metrics. Other important considerations include the desired level of accuracy, the complexity of the property to be characterized, and the available computational resources. Among all stages of the design process, the determination of lattice properties is often the most time-consuming step. However, by employing efficient evaluation methods, the overall design workflow can be significantly accelerated. This section provides an review of the main approaches used to evaluate the properties of lattice structures, highlighting their advantages and limitations. Two main approaches exist to evaluate the properties of lattice structures: single-scale modeling and multiscale modeling.

2.2.1 Single-Scale Modeling Approaches

Single-scale modeling approaches focus on the evaluation of the effective properties of lattice structures at a macroscopic level. Tree methods are commonly used: direct numerical methods, simplified models, and mesh-free methods, as illustrated in Figure 2.6.

Direct Numerical Methods

Direct numerical simulations aim to resolve the complete structure without simplifications, typically using the Finite Element Method (FEM). This approach is particularly well suited for complex geometries, as it can accurately capture local deformations and stress distributions within the lattice. Many studies have been conducted to evaluate the

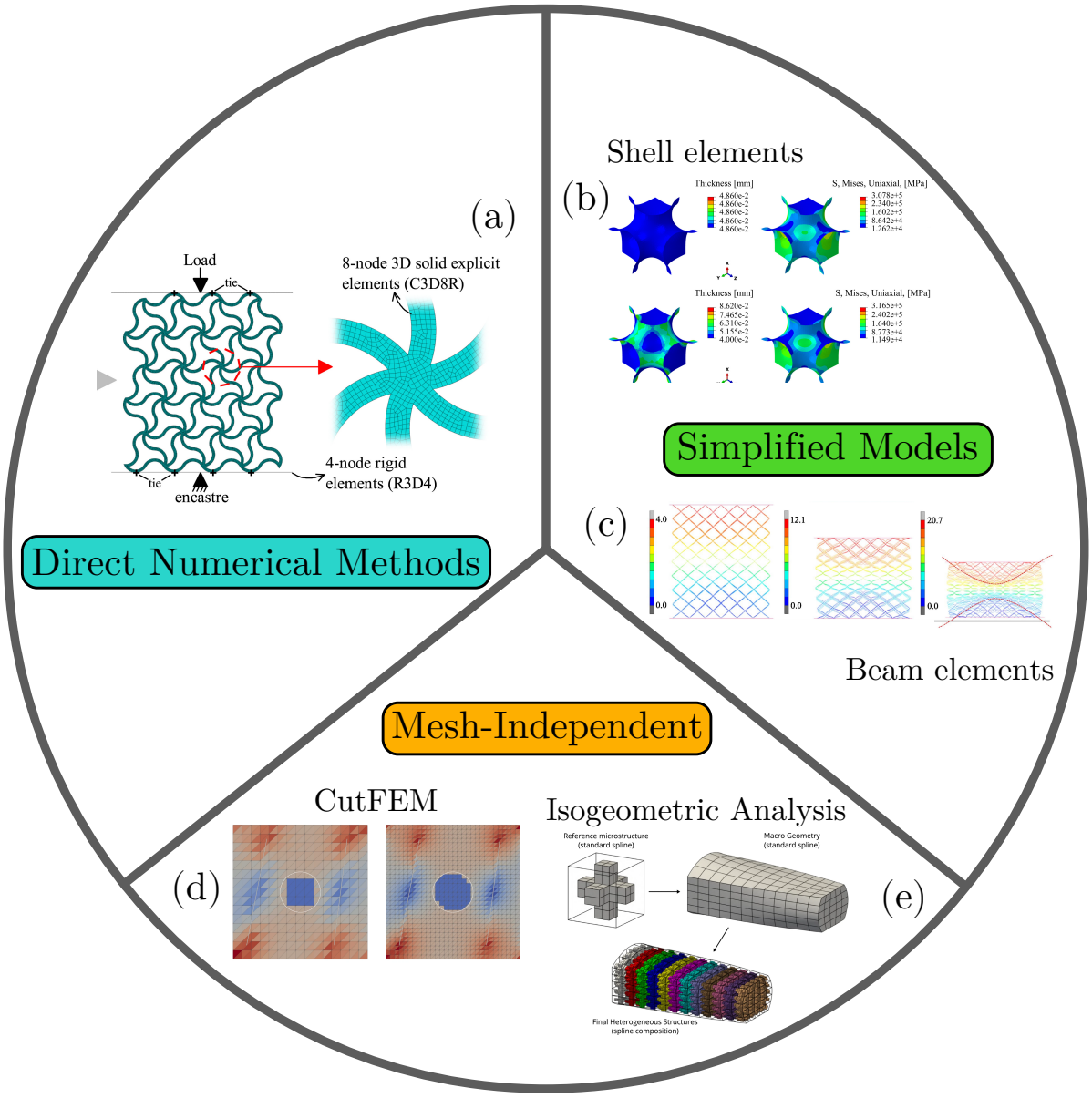

Fig. 2.6 Overview of single-scale modeling approaches used for the evaluation of properties of lattice structures. Adapted with permission from (a) (Y. Zhang, M. Li, et al. 2025) (b) (Q. Ma et al. 2021) (c) (H. Guo et al. 2020) (d) (Kerfriden et al. 2020) (e) (Hirschler, Antolin, et al. 2022).

mechanical properties of lattice structures using the FEM and to compare them with experimental results (D. Liu et al. 2025; Yousefi et al. 2022). K. Zhong et al. (2025) use the FEM to investigate the mechanical behavior of bimaterial ceramic/polymer lattice structures and demonstrate that it enhances the SEA of the structure. T. Sun et al. (2025) employ 3D solid FEM to study the energy absorption performance of dual bio-inspired structures combining turtle shell support and honeycomb architecture. Y. Zhang, M. Li, et al. (2025) apply 3D solid FEM to analyze the nonlinear mechanical behavior of horseshoe-inspired lattice structures with multiple geometric parameters, showing that the failure mode can transition from bending to stretching by adjusting the cell geometry angle. In the case of two-dimensional lattice structures, J. Ma et al. (2025) employed plane-stress FEM to investigate auxetic structures inspired by deep-sea sponges.

The main advantage of direct numerical methods lies in their high level of accuracy. However, they are computationally expensive, particularly for large structures, complex material behaviors, or internal contact problems. Furthermore, generating a conformal mesh that accurately represents the complex geometry can be challenging. While this method is well suited for evaluating the effective properties of specific lattice structures, it is not scalable for the optimization of large design spaces.

Simplified Models

Simplified models aim to provide a rapid evaluation of the effective properties of lattice structures. By posing hypotheses on the geometry and the material behavior, these models can significantly reduce the computational cost associated with the evaluation of effective properties. Methods can be divided into two categories: the analytical models and the simplified numerical models.

A well-known example of analytical models is the Gibson-Ashby model (L. J. Gibson and M. F. Ashby 1997), which relates the effective properties of truss-based lattice structures to their relative density and geometry. This model derives scaling laws for mechanical properties such as stiffness and strength, based on idealized assumptions where the lattice elements deform predominantly by bending. It provides simple closed-form relationships, typically showing that the effective Young's modulus and strength scale with the relative density raised to a power depending on the deformation mechanism. Several studies have extended the Gibson-Ashby model to account for more complex structural behaviors (H. Zhong, T. Song, et al. 2023) or to better capture the response of low relative density geometries (H. Zhong, Das, et al. 2023). Different lattice geometries, such as TPMS, have also been studied analytically. For instance, Dadashi and Rahimi

(2025) developed an analytical model to predict the combined shear and bending effects in Primitive unit cells. Analytical models are particularly useful for rapid screening of design parameters and for understanding the fundamental relationships between geometry and effective properties. However, they are limited by their assumptions and may not accurately capture the behavior of complex or anisotropic lattice structures.

Another approach to find the properties of full-scale models is to use simplified numerical models. Particular geometries can be approximated using simplified representations of their structural features. For example, TPMS structures can be modeled as shell elements (Fu et al. n.d.; H. Liu, W. W. S. Ma, et al. 2024; Q. Ma et al. 2021), reducing the computational cost of FEM. Another example is truss-based lattices, which can be modeled as beam elements (Boursier Niutta et al. 2022; Gümrük and Mines 2013; H. Guo et al. 2020; Labeas and Sunaric 2010; Luxner et al. 2005; Smith et al. 2013; Tahmasebimoradi et al. 2021). Simplified numerical models are particularly effective for structures with a high number of elements, as it significantly reduce the number of degrees of freedom in the model. However, simplified beam models rely on several assumptions that must be considered. Standard beam models may overlook material concentration at junctions within lattice structures (see Figure 2.7(a)), which leads to underestimating the stiffness of the overall structure (Gümrük and Mines 2013; H. Guo et al. 2020; Labeas and Sunaric 2010; Luxner et al. 2005; Smith et al. 2013; Tahmasebimoradi et al. 2021). To address this limitation, several strategies have been proposed to modify the stiffness at the junctions (Gümrük and Mines 2013; H. Guo et al. 2020; Labeas and Sunaric 2010; Luxner et al. 2005; McDonnell et al. 2024; Smith et al. 2013; Tahmasebimoradi et al. 2021; Terriault and Brailovski 2018). However, no consensus on the methods to account for the lack of rigidity at the nodes has been found in the literature. Therefore, chapter 3 will present a method to find the optimal penalty parameters for the joint stiffening in the context of static, dynamic, and nonlinear analyses (Cadart, Hirschler, Bahi, et al. 2025) (see Figure 2.7(b)).

Mesh-Independent Simulation Approaches

Conventional FEM methods require a conformal mesh that precisely follows the boundaries of the structure. While effective for simple geometries, this constraint poses major challenges when dealing with highly complex, graded, or interpenetrating lattice structures. To address this issue, several mesh-independent simulation approaches have been developed, enabling computations without requiring the mesh to align with the physical boundaries. These methods often rely on implicit representations of the geometry,

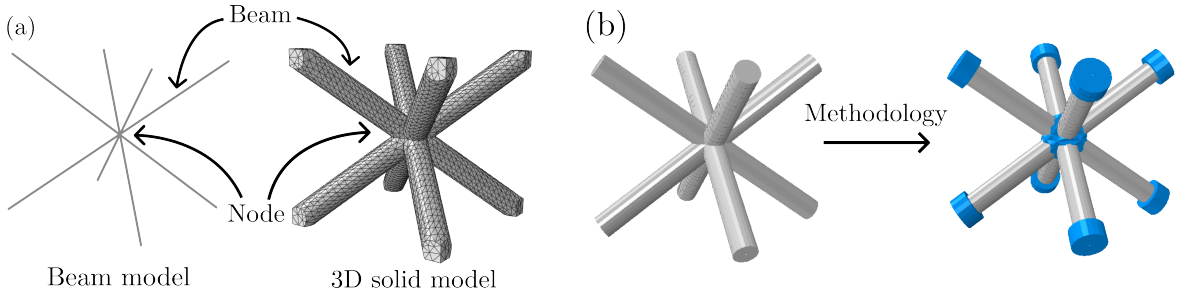

Fig. 2.7 (a) Comparison of node dimensions according to the beam model and the 3D model for the Body-Centered Cubic (BCC) (b) Methodology applied to the BCC unit cell. Reproduced with permission from (Cadart, Hirschler, Bahi, et al. 2025).

such as level-set functions, and incorporate boundary effects directly into the numerical formulation.

Embedded domain methods, such as the Cut Finite Element Method (CutFEM) (Burman et al. 2015) and the Immersed Boundary Method (IBM) (Kim and Choi 2019), allow simulations to be performed on simple background meshes while accurately capturing the influence of complex internal boundaries. In CutFEM, the physical domain is embedded within a regular background mesh. Elements intersected by the boundary are treated with special integration schemes to maintain accuracy. To ensure numerical stability, additional stabilization techniques such as ghost penalty methods are introduced, controlling the conditioning even when elements are only partially intersected. This strategy enables high-fidelity simulations of intricate geometries without the need for complex mesh generation, while preserving the convergence properties of classical FEM. In IBM, the system is immersed in a Cartesian mesh, and its effect on the governing equations is introduced through modifications near the interface. Originally developed for fluid-structure interactions, IBM has been successfully adapted to solid mechanics problems, offering a flexible framework when conformal meshing is impractical. Both CutFEM and IBM methods eliminate the need for remeshing when the geometry evolves, making them highly attractive for the simulation of architected materials and complex lattice structures.

Isogeometric Analysis (IGA) offers an alternative approach that integrates geometric modeling and numerical simulation within a unified framework (Nguyen et al. 2015). Based on NURBS or B-Splines, IGA maintains an exact representation of complex and smooth geometries throughout the analysis process. By eliminating geometric discretization errors, IGA improves simulation accuracy and robustness, especially for problems involving smooth boundary variations. Combined with implicit modeling, these mesh-independent methods efficiently streamline the simulation of lattice structures by

reducing preprocessing effort, avoiding remeshing, and ensuring high-fidelity representation of complex or graded topologies. However, despite their numerous advantages, these methods are generally more complex to implement than classical FEM (Guerder et al. 2023). They often require specific integration schemes, stabilization techniques, or adapted solvers to correctly handle the interaction between the geometry and the background mesh.

Nevertheless, several studies have successfully employed these mesh-independent methods to simulate complex lattice structures. Mikaeili et al. (2022) developed a mixed concurrent multiscale method based on CutFEM facilitating the simulation of heterogeneous materials with complex microstructures; their approach has been successfully applied to the simulation of trabecular bone from micro-CT medical images. Kerfriden et al. (2020) used a mixed-dimensional CutFEM approach to simulate fiber-reinforced composites, enabling the accurate representation of complex fiber geometries and their interactions with the matrix material. Similarly, Hirschler, Antolin, et al. (2022) and Hirschler, Bouclier, et al. (2024) employed IGA to simulate lattice structures efficiently, developing fast multiscale assembly methods that reduce computational cost while preserving high accuracy.

2.2.2 Multiscale and Decomposition Strategies

An alternative strategy is to exploit their inherent scale separation through multiscale modeling approaches or by decomposing the problem into smaller, more manageable subproblems. Multiscale modeling aims to describe the mechanical behavior of materials across several spatial scales, from the microstructure up to the macroscopic response. Each scale is treated independently, and information is transferred between them to obtain the effective behavior of the global structure. In decomposition methods, the global problem is divided into smaller subdomains that can be solved independently, often in parallel, and then recombined to obtain the overall solution. Several methodologies have been developed to efficiently address the quasi-periodic nature of lattice structures, as illustrated in Figure 2.8.

Multiscale Methods

Concurrent multiscale approaches have been introduced to enrich macroscopic models with detailed microscale information. Among them, multilevel FEM, often referred to as Two-scale Finite Element (FE^2) (Glaesener, Bastek, et al. 2021; Monaldo and Marfia 2021; Raju et al. 2021), couples a microscopic Finite Element (FE) model at each integration

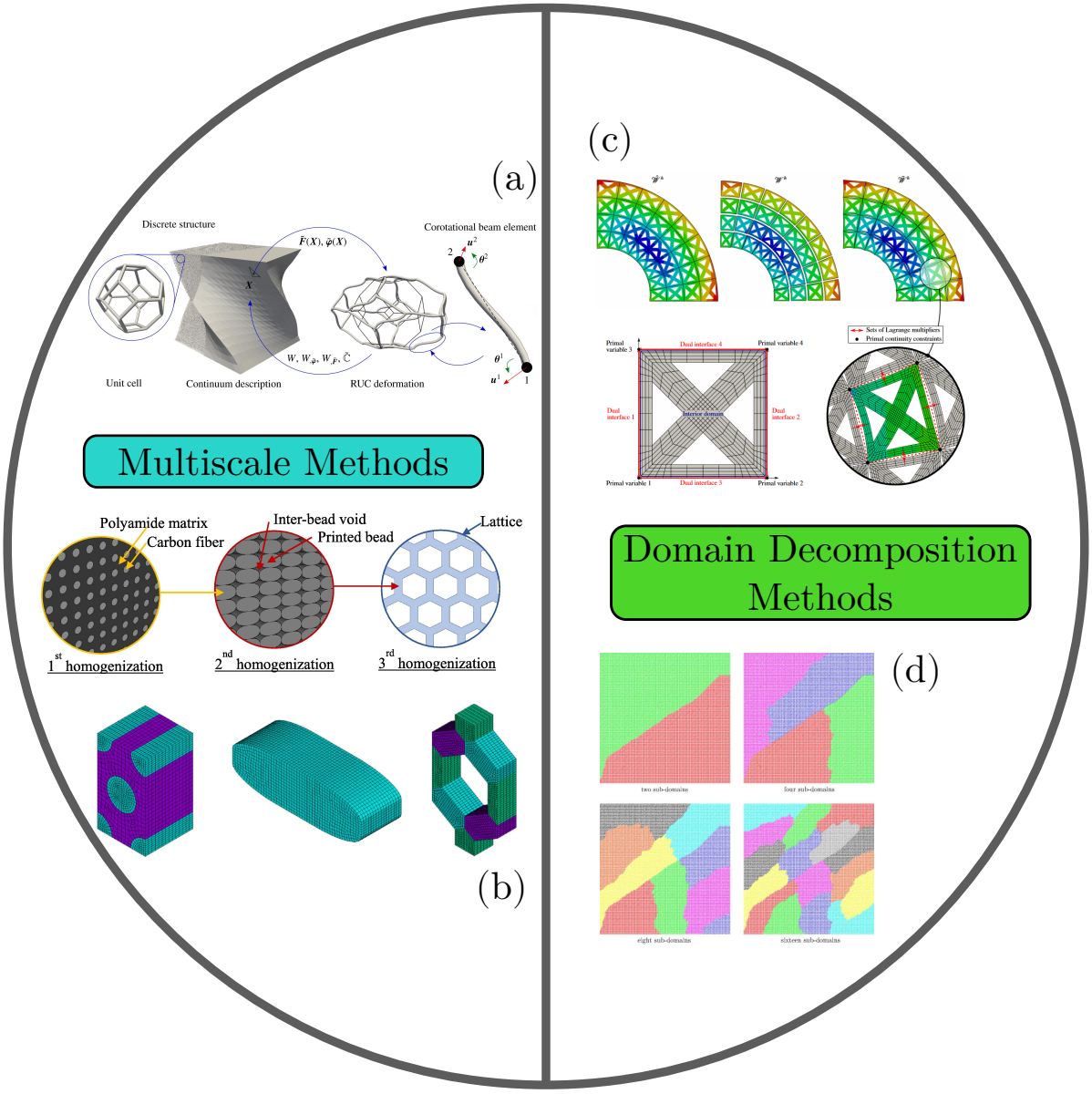

Fig. 2.8 Overview of multiscale and decomposition strategies used for the evaluation of lattice structure properties. Adapted with permission from (a) (Glaesener, Träff, et al. 2020) (b) (Marchal et al. 2024) (c) (Hirschler, Bouclier, et al. 2024) (d) (Maday and Magoulès 2006).

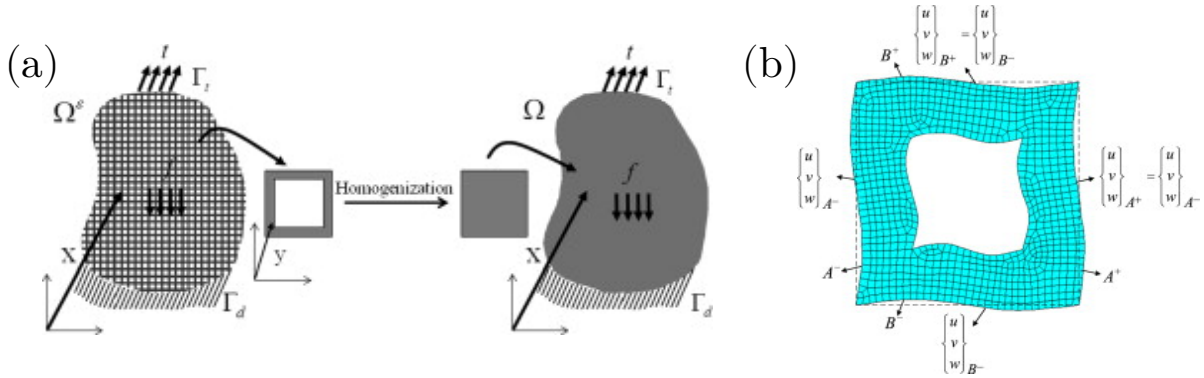

Fig. 2.9 (a) Representation of the homogenization process (b) Periodic boundary conditions applied to opposite faces of the RVE. Adapted with permission from (Arabnejad and Pasini 2013; Phani and Hussein 2017).

point of the macroscopic FE model to evaluate the homogenized response, making it particularly suitable for capturing nonlinear or anisotropic effects. Alternatively, the Multiscale FEM (MsFEM) (Henyš et al. 2019; J. Wang, J. Zhu, et al. 2023; H.-w. Zhang et al. 2010) constructs enriched basis functions that embed microscale heterogeneity directly into the macroscopic problem, offering a more efficient compromise between accuracy and computational cost. In contrast to these concurrent strategies, homogenization techniques adopt a more classical approach in which the microscale response is precomputed once on a Representative Volume Element (RVE) and then used to derive effective material properties at the macroscale (Glaesener, Lestringant, et al. 2019; Phani and Hussein 2017; Somnic and Jo 2022b). This strategy is particularly attractive for large-scale simulations, as it enables a significant reduction in computational costs while maintaining accuracy at the macroscopic level.

Homogenization theory Homogenization theory offers an efficient way to model complex periodic microstructures by replacing a heterogeneous domain Ω^ε, composed of a repeated unit cell pattern, with an equivalent homogeneous medium Ω that captures its macroscopic behavior. This transformation is schematically illustrated in Figure 2.9(a), where the detailed microstructure and boundary conditions of the fine-scale model are replaced by an equivalent continuous domain governed by effective properties. A rigorous mathematical formulation of this procedure is provided by Asymptotic Homogenization (AH), which assumes that the solution fields (e.g., displacement) depend on two distinct spatial variables: a slow (macroscale) variable $\mathbf{x}$ and a fast (microscale) variable $\mathbf{y} = \mathbf{x}/\varepsilon$, where $\varepsilon \ll 1$ is a small parameter representing the scale separation. By expanding the displacement field $u^\varepsilon(\mathbf{x})$ as a power series in ε, and substituting this expansion into the

governing equations of elasticity, one obtains a set of nested problems at each order of ε. Solving the first-order problem over a periodic unit cell Ω_0 with Periodic Boundary Conditions (PBCs) enables the computation of a fluctuating field that, when averaged, yields the homogenized stiffness tensor $\mathbb{C}^H$ of the equivalent material (see Figure 2.9 (b)). This technique ensures satisfaction of the Hill-Mandel condition and allows the effective properties to be determined without requiring full-resolution simulation of the entire structure. In practice, this means that the effective stiffness tensor $\mathbb{C}^H$ can be computed by solving a local elasticity problem on the RVE, subjected to PBCs and unit strain loadings, and then averaging the resulting stress fields over the volume of the unit cell:

$$\mathbb{C}^H : \boldsymbol{E} = \frac{1}{|\Omega_0|}\int_\Omega \boldsymbol{\sigma}(\mathbf{y})d\mathbf{y} = \frac{1}{|\Omega_0|}\int_\Omega \mathbb{C}(\mathbf{y}) : \left(\boldsymbol{E} + \boldsymbol{\varepsilon}(\boldsymbol{\chi}^{(k)}(\mathbf{y}))\right) d\mathbf{y} \tag{2.6}$$

where $\boldsymbol{E}$ is the imposed macroscopic strain, $\boldsymbol{\chi}^{(k)}$ are the periodic corrector functions associated with each component of $\boldsymbol{E}$, and $\boldsymbol{\sigma}$ is the local stress field within the RVE. A fundamental assumption, however, is the existence of a clear scale separation, meaning that the characteristic size of the cell must be significantly smaller than the overall dimensions of the structure.

Extensive research has focused on the application of homogenization techniques to lattice structures (Arabnejad and Pasini 2013; Glaesener, Träff, et al. 2020; Hassani and Hinton 1998; Savio et al. 2019; Somnic and Jo 2022a). A notable example is the recent work of Marchal et al. (2024), which proposes a three-level homogenization strategy for predicting the effective behavior of hierarchical composite honeycomb lattices. In Arabnejad and Pasini (2013), this method was successfully applied to a wide range of lattice topologies, yielding accurate predictions of stiffness and strength over various relative densities, and serving as a benchmark for evaluating alternative homogenization schemes. Nowadays, in this context of multiscale homogenization, it is a common practice to use the microscale models to train surrogate models which are then used later on during the macroscale simulations to reduce the overall computational time (polynomial function (Costa et al. 2022), kriging (X. Liu et al. 2022), neural network (Dos Reis and Karathanasopoulos 2022; Z. Gao et al. 2023; Ross and Hambleton 2021)), more information in section 2.4. Nonetheless, the assumption of a clear scale separation is not always valid, particularly when the geometry varies from one cell to another, as is frequently the case in graded or multifunctional architectures (J. Wu, Sigmund, et al. 2021). Moreover, the hypothesis of scale separation is often violated due to the limitations of AM processes, which are typically unable to fabricate structural features below a certain resolution threshold. Some advanced homogenization methods aim to relax the scale

separation hypothesis, but this comes at the cost of significantly increased computational complexity (Somnic and Jo 2022b; Yvonnet and Bonnet 2014). Alternatively, the standard homogenization approaches are sometimes complemented with post-processing steps, in which full-scale simulations are performed to assess the discrepancies introduced by the homogenization assumptions (Groen and Sigmund 2018). Furthermore, for complex material behaviors such as nonlinear elasticity, plasticity, damage, or time-dependent responses, multiscale approaches often require demanding and computationally intensive procedures, for instance the FE^2 method (Raju et al. 2021), as previously mentioned.

These techniques are well-suited for periodic media but may become inaccurate when the assumption of periodicity or scale separation breaks down, as frequently encountered in complex lattice structures.

Domain Decomposition Methods

Domain decomposition methods constitute an alternative approach to multiscale modeling, characterized by splitting the global problem into smaller subproblems associated with distinct sub-domains. Each local problem is solved independently, and equilibrium and compatibility are enforced at the interfaces between neighboring sub-domains. In contrast to homogenization techniques, these methods do not rely on a clear scale separation, which makes them particularly suitable for structures exhibiting complex geometries or weakly separated length scales. Moreover, they are inherently well adapted to parallel computing, since the resolution of sub-domain problems can be performed simultaneously.

Within this framework, two main classes can be distinguished: primal and dual approaches (Gosselet and Rey 2006). Primal methods, such as the Balanced Domain Decomposition (BDD) approach, consider the displacements (or primal variables) on the interfaces between sub-domains as the main unknowns (Mandel 1993; Toselli and Widlund 2005). In contrast, dual methods, like the Finite Element Tearing and Interconnecting (FETI) method, formulate the problem in terms of interface forces (Lagrange multipliers), which act as dual variables enforcing compatibility between sub-domains (Farhat and Roux 1991; Hirschler, Bouclier, et al. 2024; Klawonn and Rheinbach 2006). Dual-primal methods combine both displacements and interface forces as unknowns and often provide improved convergence in complex configurations (Dohrmann 2003; Farhat, Lesoinne, et al. 2000; Mandel et al. 2005).

From a geometric perspective, domain decomposition methods can also be classified according to the configuration of the sub-domains: overlapping and non-overlapping approaches. Non-overlapping methods preserve the strict physical partitioning of the domain, and require explicit interface conditions—either through direct coupling or via

Lagrange multipliers—to guarantee equilibrium and continuity across sub-domains. In contrast, overlapping formulations extend each sub-domain into neighboring regions, thereby enhancing the robustness of the solution process by enabling the exchange of redundant information across interfaces.

Lattice structures are particularly well suited for domain decomposition methods, as their cellular nature allows for a natural partitioning into non-overlapping sub-domains corresponding to individual unit cells or groups of cells. This property has motivated the development of tailored solvers that exploit the repetitive geometry and mechanical behavior of such systems. For instance, Hirschler, Bouclier, et al. (2024) propose a reduced-order modeling-based inexact FETI-Dual-Primal (FETI-DP) solver specifically adapted to lattice structures. By integrating model reduction techniques into a dual-primal non-overlapping framework, their method takes advantage of the geometric similarity among unit cells to significantly reduce memory requirements and computational cost, while maintaining high accuracy and scalability. Beyond structural analysis, domain decomposition strategies have also been successfully employed in the context of bandgap optimization for lattices (W. Guo et al. 2025; Vouvakis et al. 2006).

Despite their potential, domain decomposition methods are not as widely employed as homogenization techniques in the context of lattice structures. Nevertheless, they constitute a promising alternative for the simulation of complex geometries and heterogeneous materials, as they do not require a strict separation of scales. In addition, their natural compatibility with parallel computing makes them particularly attractive for large-scale high-fidelity simulations.

2.2.3 Experimental Characterization

Experimental characterization remains the most reliable approach for determining the physical properties of lattice structures, especially when dealing with ultra-complex geometries or non-linear material behaviors. Although essential, this phase is generally costly and time-consuming, as it requires the fabrication and testing of numerous physical samples. In practice, experimental results are often used as a reference to validate numerical models, ensuring that simulations accurately capture the mechanical response of the lattice structures under consideration. The methods are highly diverse and have been applied to a wide range of behaviors, including mechanical, thermal, electrical, and fluid phenomena. Different experimental techniques can be employed to characterize the mechanical response of lattice structures. Compression tests, illustrated in Figure 2.10(a), are the most commonly used, as they provide a straightforward means of evaluating mechanical properties and are particularly relevant for assessing energy

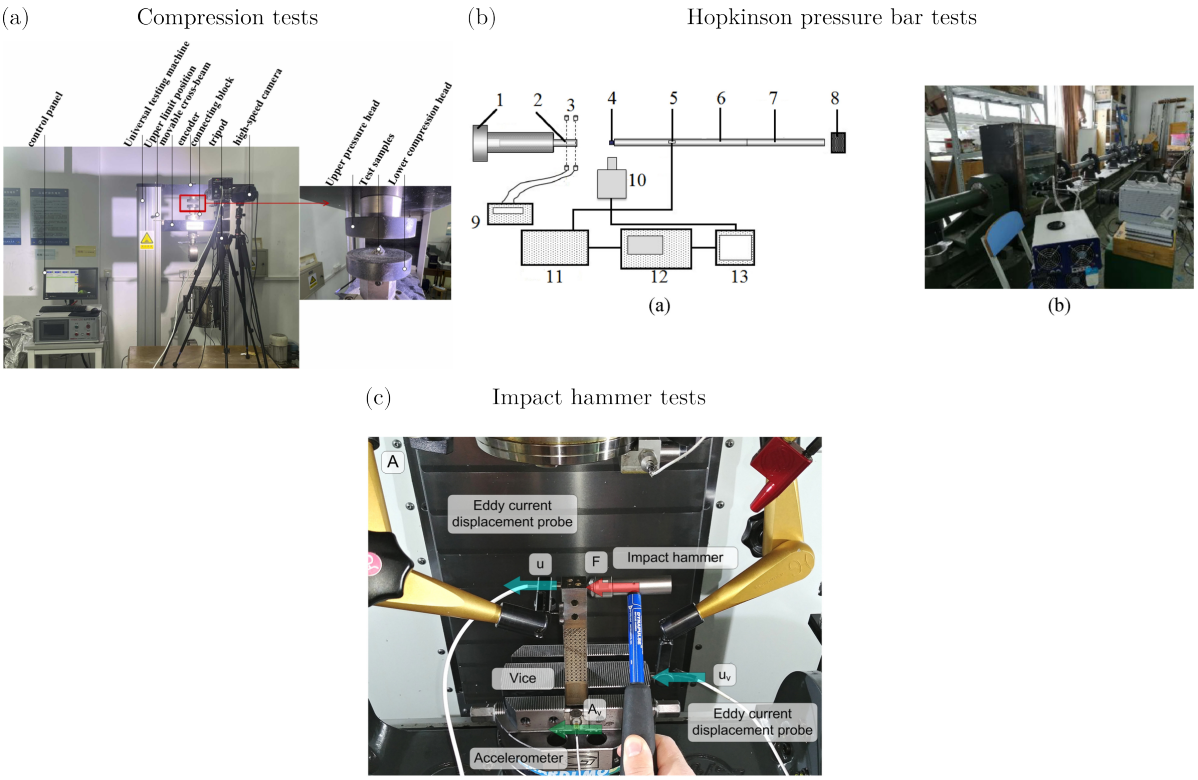

Fig. 2.10 Overview of experimental characterization techniques. Adapted with permission from (a) (D. Liu et al. 2025) (b) (Xiao et al. 2020) (c) (Scalzo et al. 2021).

absorption capabilities (Ling et al. 2019; D. Liu et al. 2025; Rodrigo et al. 2023). At higher strain rates, impact experiments such as Hopkinson pressure bar tests are often performed to investigate the dynamic response of lattices under extreme loading conditions (Figure 2.10(b)) (X. Wang, Qin, et al. 2022; Xiao et al. 2020). For frequency-domain characterization, pulse testing with an impact hammer or shaker has also been employed to identify the dynamic properties of lattice structures (Figure 2.10(c)) (Ozdemir et al. 2023; Scalzo et al. 2021). Finally, while compressive and tensile behaviors often differ markedly, tensile tests—though less common—remain useful, especially for soft materials or when evaluating the tensile strength of lattice structures (Q. Hu et al. 2023; Jung et al. 2023).

Lattice structures exhibit both local and global deformations and failures, depending on the loading conditions, the geometry of the structure, and the material properties. Some simulation methods, such as homogenization techniques, can not capture this local–global behavior, which calls for the development of dedicated solvers capable of resolving both scales consistently. For instance, Damanpack et al. (2019) developed a

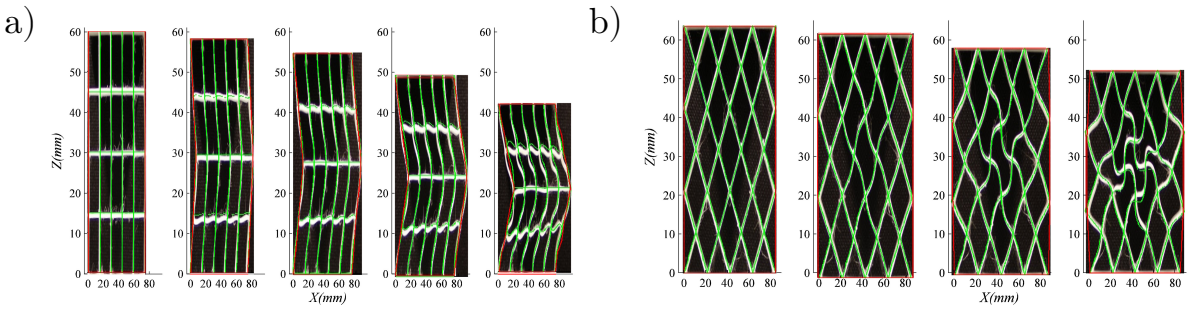

Fig. 2.11 Compressive loading with two differents lattice geometries showing the local/-global behavior dependence (white lines: experiment, green lines: large-scale modeling, red border: multiscale modeling). Adapted with permission from (Damanpack et al. 2019).

multiscale model to predict the mechanical response of a hyperelastic lattice structure (see Figure 2.11).

The biggest source of differences between the numerical and experimental results is the manufacturing process. Depending of the AM process defects, porosity, and other geometric variations can appear and significantly affect the mechanical properties of the final structure. To address this issue, several studies have employed advanced imaging techniques to characterize the microstructure and evaluate the defects in lattice structures (Sadeghi et al. 2020; Xiao et al. 2020). W. Wu et al. (2020) studies go even further by using synchrotron X-ray micro-computed tomography imaging during compression tests allowing to observe the effect of manufacturing defects on the mechanical behavior of lattice structures (W. Wu et al. 2020). Advancing our understanding of the mechanical behavior of printed structures will ultimately enable us to exploit the full potential of lattice architectures, through precise control of their geometry and material properties to achieve targeted performance.

Concluding Remarks on Property Evaluation

The evaluation of the effective properties of lattice structures is a critical step in understanding their physical behavior. The advantages and limitations of each evaluation methodology must be carefully assessed in order to select the most appropriate approach for the intended application. Direct numerical simulations provide high-fidelity results but are computationally expensive and difficult to scale for design optimization. Simplified models, including analytical and beam-based methods, offer faster evaluations at the cost of reduced accuracy. Mesh-independent techniques like CutFEM or IGA provide an elegant solution to handle complex and evolving geometries, although their implementa-

tion complexity remains a challenge. Multiscale methods, particularly homogenization, are powerful tools when scale separation exists, but become unreliable in the presence of gradients or strong geometrical heterogeneity. Domain decomposition methods, on the other hand, do not rely on this assumption and offer a scalable and flexible alternative for the simulation of architected materials. Finally, experimental characterization remains essential for validating numerical predictions and understanding the real mechanical behavior of printed structures, including the impact of manufacturing defects. A rigorous evaluation strategy should thus combine complementary approaches, leveraging the speed of reduced models, the accuracy of high-fidelity simulations, and the validation power of experiments, in order to guide the design of high-performance lattice structures.

2.3 Optimal Design for Mechanical Lattice Structures

Finding the optimal design of a mechanical structure is a challenging and computationally intensive task that requires appropriate methodologies tailored to the nature of the system and the type of parameters to be optimized. Depending on the application context, the optimization process can target different scales and features: material properties, geometric dimensions, global shape, structural topology, or even manufacturing strategies. In structural design, three main classes of optimization strategies are commonly distinguished (J. Gao et al. 2020; Tyflopoulos and Steinert 2020) (see Figure 2.12):

- **Size optimization**, where the objective is to determine the optimal values of a set of predefined geometric or physical parameters (e.g., thickness, angle, radius, material).
- **Shape optimization**, which focuses on modifying the boundary of the structure to improve its performance while keeping the topology fixed.
- **Topology Optimization (TO)**, which explores the distribution of material within a given design domain to identify the most efficient load paths and possibly remove redundant regions (J. Zhu et al. 2021).

These three strategies are not mutually exclusive and can be combined depending on the problem formulation and design objectives. In the case of lattice structures, the design optimization problem becomes even more complex due to their inherently quasi-periodic nature. The optimization process can be carried out at different hierarchical levels: the cell-scale (e.g., material or geometry of the unit cell), the structure-scale

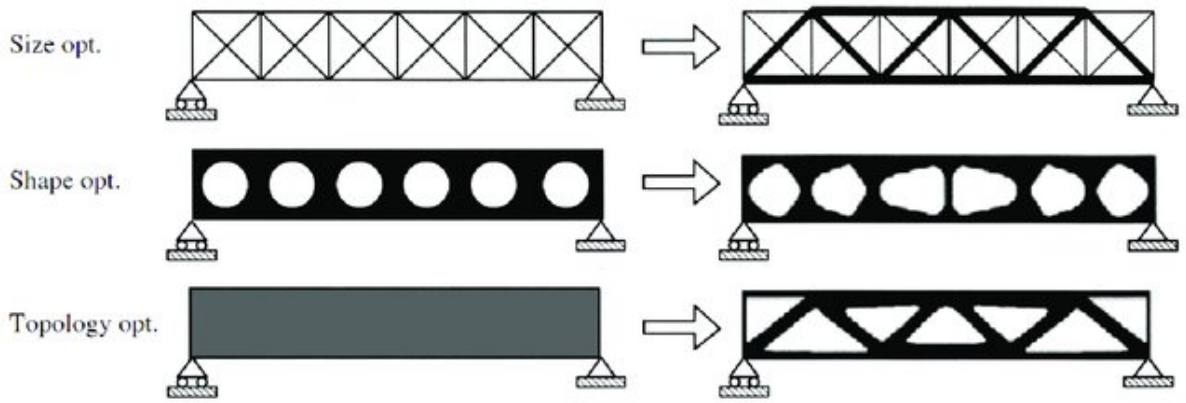

Fig. 2.12 Optimization methods for structures. Adapted with permission from (Gebisa and Lemu 2017).

(e.g., distribution and orientation of unit cells or gradient over the structure), or any combination thereof. Depending on the level considered, different optimization strategies may be more appropriate, as the design variables, constraints, and objectives can differ significantly between scales. The coexistence and interaction of these scales in lattice structures render the optimization task highly complex and computationally intensive (Pan et al. 2020; J. Wu, W. Wang, et al. 2021).

In this section, we review the main optimization strategies used in the literature for lattice structures. These approaches are categorized according to the scale at which the optimization is conducted: cell-scale, structure-scale, or concurrently at both levels (see Figure 2.13). In addition to these scale-specific strategies, a separate class of optimization methods known as metaheuristic algorithms has also been widely used. These approaches are not restricted to a specific scale and can be applied across different levels of the design process. They are particularly well suited for exploring high-dimensional, non-convex, or multi-objective design spaces where gradient-based methods may fail.

2.3.1 Optimization Strategies at the Cell-Scale

Due to the periodic nature of lattice structures, the mechanical response of individual unit cells has a significant impact on the global behavior of the entire structure. At the cell-scale, a wide variety of parameters can be optimized, ranging from the relative density of the unit cell to its internal multimaterial topology (Gómez-Silva et al. 2025; Kazemi et al. 2020). Extensive research has primarily focused on the optimization of simple geometric parameters such as the cross-sectional shape and size (X. Wang, L. Zhu, et al. 2021; M. Zhao, D. Z. Zhang, et al. 2022). Shape optimization has also been explored,

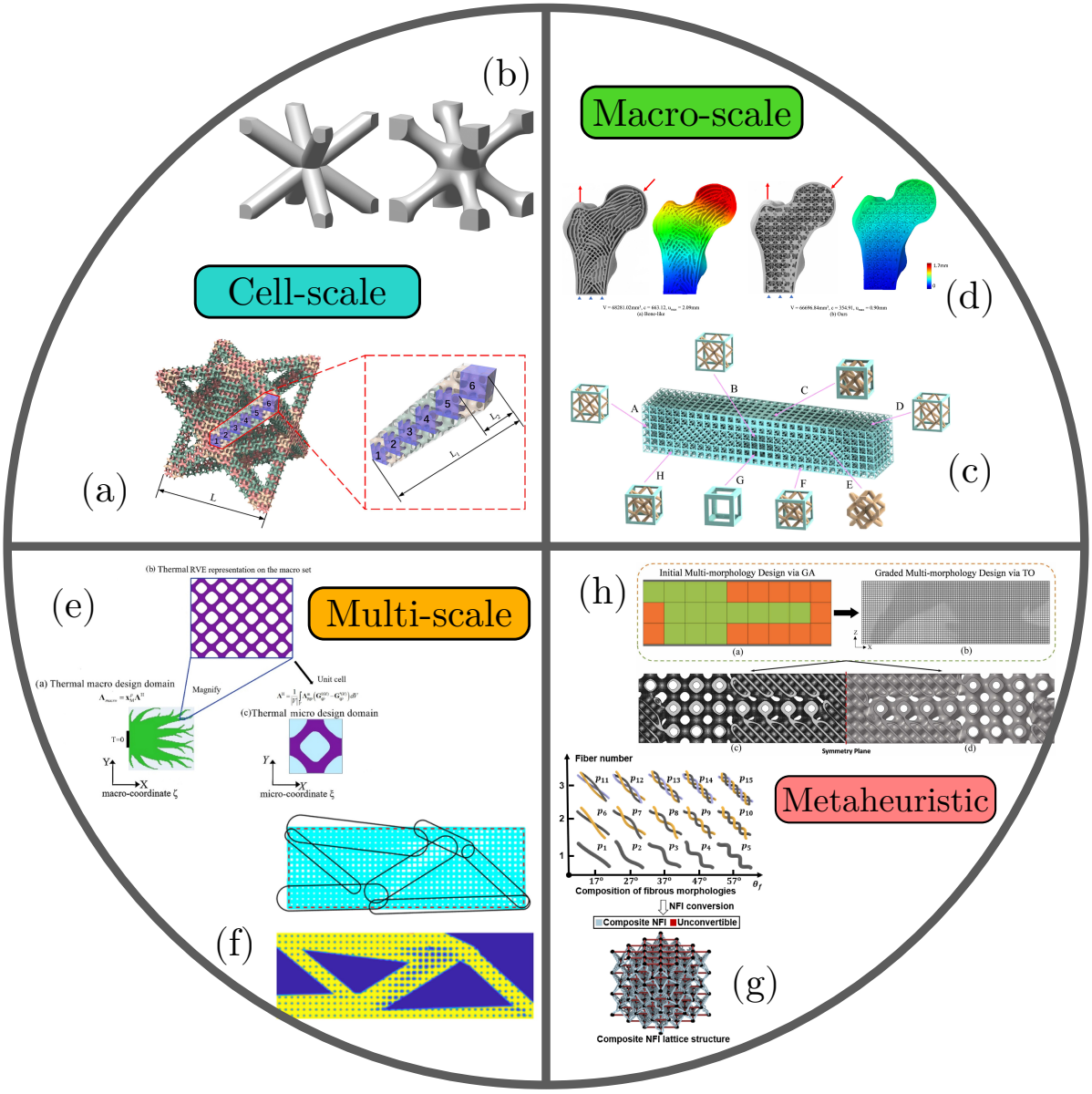

Fig. 2.13 Overview of optimization methods for lattice structures. Adapted with permission from (a) (Yin, F. Meng, et al. 2023) (b) (Ali and Abdi 2023) (c) (X. Wang, Qin, et al. 2022) (d) (Y. Liu, P. Zhang, et al. 2025) (e) (Al Ali and Shimoda 2024) (f) (Hoang, Tran, et al. 2020) (g) (Z. Gao et al. 2023) (h) (Ozdemir et al. 2023).

notably in the work of Ali and Abdi (2023), where a method is proposed to smooth both the geometry and the density distribution of a BCC unit cell. In Yin, F. Meng, et al. (2023), a hierarchical unit cell combining an octet-truss with TPMS is designed to enhance the SEA, and a surrogate model is developed to predict this property based on geometric inputs. Kazemi et al. (2020) introduce a TO framework for multimaterial truss-based lattice structures, enabling the optimization of both the topology and the material distribution within the unit cell.

Optimizing the properties of the unit cell is crucial for enhancing the overall performance of the lattice structure. Once optimized, the unit cell is periodically repeated to construct the entire structure. However, this periodic repetition limits the applicability of such designs for multifunctional components, where local mechanical requirements vary and necessitate spatially graded unit cell properties to adapt to non-uniform loading conditions.

2.3.2 Optimization Strategies at the Macro-Scale

Using different cell geometries in the same structure is a common approach to optimize the overall performance of lattice structures and to adapt to local loading conditions. However, the number of parameters to optimize increases significantly, as each unit cell can be assigned different geometric parameters. Moreover, the simulation of the entire structure becomes more computationally demanding, as it requires high-resolution models to accurately capture the geometric and mechanical complexity of the structure.

To address these challenges, several studies have proposed predefined spatial distributions of one or more unit cell types across the structure, in order to reduce the computational cost associated with determining the optimal spatial arrangement (D. Li, Qin, J. Xu, et al. 2022; Musenich et al. 2023). Since these methods adopt a global perspective on the structure, the notion of distinct unit cells is often relaxed, which makes them particularly well-suited for the design of conformal lattice structures. Two main types of approaches have been developed to address the optimization of such conformal structures:

- Global modeling of the entire lattice followed by direct optimization (Y. Liu, P. Zhang, et al. 2025; Stankovic et al. 2015; Vaissier et al. 2019),
- Global optimization under specific constraints and parameters explicitly designed for conformal lattice architectures (Stollberg et al. 2025; J. Wu, W. Wang, et al. 2021).

To facilitate the simulation process, another widely adopted strategy in the literature involves homogenization-based techniques, which exploit the intrinsic multiscale nature of lattice structures. In this strategy, the effective mechanical properties of a unit cell are first computed through homogenization, and then incorporated into a global optimization framework such as Topology Optimization (TO). Within the TO context, the classical stiffness matrix is replaced by a homogenized stiffness tensor representing the response of the unit cell, which is then used to assemble the global stiffness matrix of the structure. Numerous studies have adopted this methodology, including (J. Chen, Baldwin, et al. 2025; Ji et al. 2024; Vijayakumaran et al. 2025; X. Wang, Qin, et al. 2022; Z. Yang et al. 2024; J. Zhang et al. 2021), demonstrating its effectiveness for concurrent optimization. Using a similar approach, X. Wang, L. Zhu, et al. (2021) applied a method of feasible directions combined with homogenization to optimize graded filleted lattice structures. This method is particularly powerful, as it enables the optimization of structures involving a large number of parameters, implicitly represented within a homogenized stiffness matrix. However, the homogenization-based approach comes with notable limitations. First, the underlying assumption of scale separation may not hold when unit cell geometries vary rapidly within the optimization domain, which compromises the validity of the homogenized model. Second, ensuring geometric and mechanical connectivity at the microscale between adjacent unit cells requires additional constraints, which increase the complexity of the optimization problem.

Macroscopic optimization of lattice structures presents both notable advantages and inherent limitations. On the one hand, it enables the spatial distribution of unit cell geometries to be tailored to non-uniform loading conditions, thus improving overall structural performance. Such flexibility allows the design of conformal lattices that better adapt to complex geometries and functional requirements, without being constrained by periodicity. Additionally, global optimization approaches may integrate multiple objectives and constraints, facilitating the generation of multifunctional components. On the other hand, the computational cost of macroscopic optimization is significant, especially when each unit cell is assigned one or multiple independent geometric parameters. The simulation of the entire lattice structure is significantly more computationally demanding than microscale approaches, as it requires high-resolution models to accurately capture the geometric and mechanical complexity of the structure. Moreover, the lack of scale separation significantly limits the applicability of homogenization techniques and complicates the computation of sensitivities required for gradient-based optimization methods.

To overcome the respective limitations of micro-scale and macro-scale optimization, recent research has focused on multiscale strategies that aim to optimize lattice structures simultaneously across multiple scales, leveraging the interactions between local and global design parameters.

2.3.3 Optimization Strategies across Multiple Scales

Optimizing both the cell-scale and the macro-scale of a lattice structure within a unified framework is a promising strategy to enhance global mechanical performance and enable multifunctionality. However, the number of design parameters increases drastically when considering both scales simultaneously, making direct optimization computationally demanding. A first class of approaches seeks to mitigate this complexity by using simplified geometric representations for the design variables (Hoang, Pham, et al. 2022; Hoang, Tran, et al. 2020). Hoang, Tran, et al. (2020) propose a TO method based on adaptive geometric primitives, using a framework composed of macro-scale moving bars and a lattice of micro-scale bars, effectively reducing the dimensionality of the optimization problem. A more advanced example is given by Y. Wu, L. Wang, et al. (2025), who first apply TO to optimize cell-level geometries and then use a deep generative model to learn the mapping between geometry and effective properties. The final step consists in applying a second TO at the macro-scale using the learned homogenized properties, completing a fully integrated multiscale optimization pipeline.

A second class of strategies adopts an iterative scheme, alternating between macro- and micro-scale optimizations. For instance, Al Ali and Shimoda (2024) proposed a two-step TO procedure for designing heat exchangers composed of periodic lattice structures, where both scales are optimized iteratively and coupled through a homogenization scheme. Santos et al. (2024) developed a two-scale concurrent optimization of the lattice structure, which involves simultaneously optimizing the topology at both the macro and micro-scales to achieve an optimal topology. To facilitate the optimization process, homogenization methods are employed to approximate the material and geometric properties of the unit cells.

These examples show that optimizing lattice structures across multiple scales using simplified models, hybrid hierarchical methods, or iterative macro and micro schemes is an effective way to enhance their performance. By exploiting the interplay between local and global design parameters, multiscale optimization approaches can yield significant improvements in structural behavior. However, the inherent complexity of such problems makes the development of efficient optimization frameworks particularly challenging. Striking a balance between computational cost and design flexibility is essential to ensure

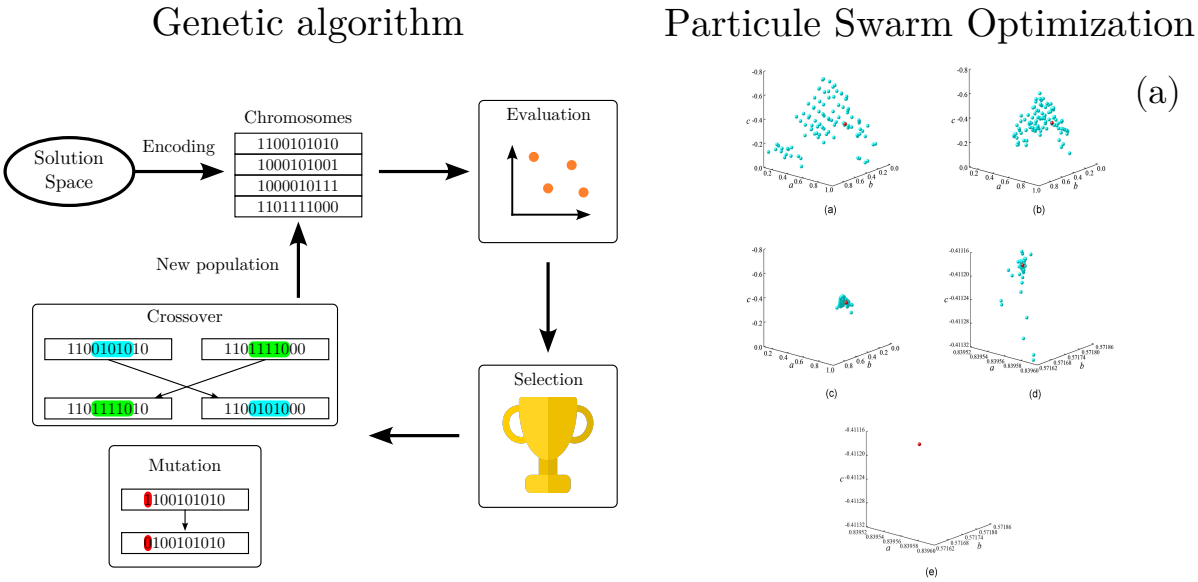

Fig. 2.14 Examples of metaheuristic algorithms. Particle Swarm Optimization (PSO) Adapted with permission from (a) (J. Zhao et al. 2025).

practical applicability. Moreover, integrating multiple optimization strategies within a single methodology further increases the complexity, which may limit their deployment in real-world engineering applications.

2.3.4 Metaheuristic Strategies

Metaheuristic optimization methods represent a broad class of algorithmic strategies inspired by natural phenomena or physical processes, designed to efficiently explore complex and high-dimensional design spaces without relying on gradient information (Velasco et al. 2024). These approaches include, among others, Genetic Algorithms (GA) (Katoch et al. 2021), Particle Swarm Optimization (PSO) (Shami et al. 2022), and Simulated Annealing (SA) (Delahaye et al. 2019), which have gained increasing attention in the design of lattice structures due to their ability to escape local minima and handle multi-objective or discrete-variable problems. Some examples of metaheuristic algorithms are illustrated in Figure 2.14. Unlike classical gradient-based methods, metaheuristics are particularly well suited for problems in which the objective function is noisy, non-differentiable, or derived from expensive simulations, as is often the case in the design of architected materials. Moreover, their population-based nature makes them inherently parallelizable and adaptable to diverse design constraints and objectives. These algorithms are particularly effective for exploring large and complex design spaces and have been applied to a wide range of optimization problems in lattice structure design.

Among metaheuristics, Genetic Algorithms (GA) are inspired by the principles of natural selection and evolution. A typical GA begins with an initial population of candidate solutions encoded as chromosomes. Through iterative application of genetic operators such as selection, crossover, and mutation, the algorithm evolves the population towards better-performing individuals according to a fitness function, illustrated in Figure 2.14. Selection favors high-quality solutions, crossover recombines features from parent individuals, and mutation introduces diversity. This evolutionary cycle continues until a stopping criterion is met, such as a predefined number of generations or convergence threshold. For example, Z. Gao et al. (2023) proposed a data-driven, coconut-inspired lattice design methodology aimed at achieving extremely recoverable and ultrahigh SEA performance. Their approach consisted in optimizing the configuration of each individual beam within the lattice structure to tailor its mechanical response precisely. Experimental results demonstrated a significant enhancement in energy absorption capabilities: an improvement factor ranging from 6 to 36 in initial SEA compared to rubber-like lattice structures. Ozdemir et al. (2023) introduced a hybrid design strategy to generate functionally graded, multi-morphology TPMS lattice structures. In a first step, the morphology and relative density of each unit cell were optimized using a GA, with the objective of minimizing the effective compliance of the structure. In a second step, a finer distribution of relative density was determined using an anisotropic homogenization-based TO method. This two-step strategy illustrated that hybrid multiscale optimization outperforms conventional single-scale approaches. Experimental validation on a graded Messerschmitt–Bölkow–Blohm (MBB) beam confirmed the benefits, with the hybrid design achieving up to a 13% increase in stiffness compared to uniform lattice configurations. Other noteworthy studies have explored the use of GA for diverse lattice optimization objectives, such as tailoring elastic properties through internal beam arrangement (Schwahofer et al. 2023), designing 2D plate-based topologies to match target homogenized mechanical responses (Dos Reis and Karathanasopoulos 2022), or optimizing support geometries in the context of AM (Vaissier et al. 2019).

The PSO algorithm is inspired by the collective behavior of swarms, such as flocks of birds or schools of fish. Each particle in the population represents a candidate solution and moves through the design space by updating its position and velocity based on both its personal best position and the global best position found by the swarm, as illustrated in Figure 2.14. These updates are guided by a balance of cognitive (individual learning) and social (collaborative learning) terms, often modulated by an inertia weight. The PSO algorithm continues iteratively until a convergence criterion is met. PSO-based methods have been used in lattice optimization, such as in (Koduru et al. 2022), and in practical

applications such as shoe sole optimization (J. Zhao et al. 2025), wind turbine structural design (J. Chen, R. Yang, et al. 2016), or slope stability assessment (Y. Liu, He, et al. n.d.).

The Simulated Annealing (SA) algorithm draws inspiration from the annealing process in metallurgy, where a material is heated and gradually cooled to achieve a stable crystalline structure. In SA, a single solution is perturbed at each iteration, and the new solution is probabilistically accepted based on the change in objective function and a temperature parameter that decreases over time. This mechanism allows the algorithm to escape local minima early on and gradually focus on exploitation as the system cools. The SA algorithm has been applied to lattice structure optimization, notably in the design of support structures for AM (White, Liang, et al. 2024; White, G. Zhang, et al. 2024), and also for TO of truss lattices (Shim and Manoochehri 1997).

Metaheuristic algorithms offer several advantages, including their robustness to local minima, flexibility in handling various types of design variables, and compatibility with black-box or non-differentiable objectives. They are particularly well suited for computational design workflows, where the design process is driven by algorithmic generation and the exploration of complex, high-dimensional parameter spaces. However, metaheuristics also present limitations: they may require extensive parameter tuning, involve large numbers of function evaluations, and may exhibit slower convergence compared to gradient-based methods.

Concluding Remarks on Optimization Strategies

In summary, the optimization of lattice structures is a complex and multifaceted task that requires careful consideration of numerous parameters and design objectives. The selection of an appropriate optimization strategy depends on the specific application context, the targeted performance metrics, and the available computational resources. A prevailing trend in recent research is to combine multiple optimization strategies across different scales within the structure, aiming to maximize performance while maintaining reasonable computational effort. Furthermore, the development of a new optimization methodology must account for all stages of the design and simulation pipeline to ensure overall coherence and efficiency. The rapid advancement of computational capabilities and algorithmic techniques has opened new avenues, particularly through the use of surrogate models that significantly reduce the computational cost of optimization. In the following section, we explore the role and potential of surrogate models in the optimization of lattice structures.

2.4 Accelerating Optimization with Surrogate Models

Surrogate models are mathematical approximations of complex functions, designed to replace computationally expensive evaluations of physical systems with more efficient alternatives. Throughout this thesis, the term will be used in this sense. They are employed in various contexts, including property evaluation, uncertainty quantification, and optimization, and can be integrated at different stages of the computational pipeline. Their main advantage lies in their ability to rapidly approximate the behavior of expensive simulations, making them particularly valuable for optimization problems that require repeated evaluations of objective functions or constraints. However, surrogate models remain approximations of the true underlying functions, and their accuracy strongly depends on the quality and diversity of the training dataset, as well as the suitability of the surrogate modeling technique to the problem at hand. Moreover, the integration level of the surrogate model within the computational workflow can significantly impact its effectiveness and applicability. In this section, we review the main surrogate modeling approaches reported in the literature in the context of lattice structures (see Figure 2.15). We then analyze their integration within the computational workflow, highlighting the associated benefits and limitations at each level of resolution.

2.4.1 Classification of Surrogate Modeling Approaches

Surrogate modeling techniques are commonly categorized into three main categories: analytical models, probabilistic models, and artificial intelligence approaches. Each category differs in terms of assumptions, data requirements, flexibility, and ability to quantify uncertainty. The following classification presents representative techniques for each category and discusses their typical applications in the context of lattice structure.

Analytical and Deterministic Surrogate Models

Deterministic surrogate models are based on the assumption that the underlying function can be approximated using a known set of basis functions. Such models are called deterministic in the sense that they produce repeatable and consistent predictions for identical input parameters, independent of any stochastic effects. These surrogate models are generally suited for low-dimensional problems. These models include:

- **Response Surface Method (RSM)** (Khuri and Mukhopadhyay 2010): A common approach involving the fitting of a polynomial function, typically linear,

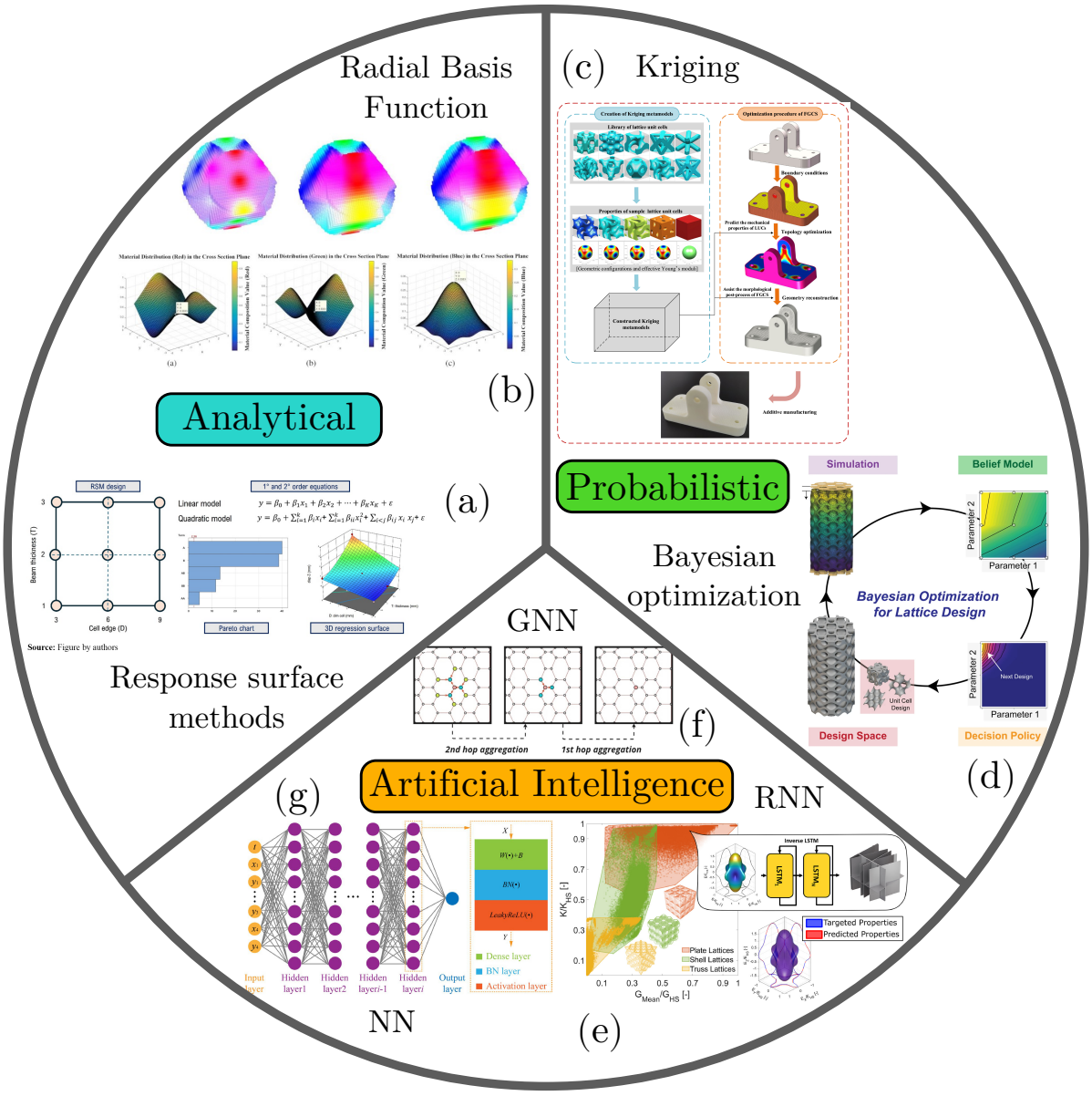

Fig. 2.15 Overview of surrogates models. Adapted with permission from (a) (Donnici et al. 2024) (b) (Y. Liu, H. Yang, et al. 2021) (c) (X. Liu et al. 2022) (d) (Gongora et al. 2024) (e) (Meyer et al. 2024) (f) (H. Wang et al. 2025) (g) (Zhou et al. 2025) .

quadratic, or higher order, to a set of data points. This method is computationally efficient and well-suited to low-dimensional problems with smooth response surfaces.

- **Radial Basis Function (RBF)** (Arora et al. 2023): These models interpolate the response by using radial functions centered at the training points. They are particularly effective for scattered data and can capture complex nonlinear behavior.

RSM has been effectively applied to predict and optimize the mechanical response of lattice structures, notably in the design of beam-based cells such as Octet and IsoTruss, where polynomial regression models were constructed to estimate stiffness as a function of cell size and beam thickness (Donnici et al. 2024) . This approach enables the identification of optimal stiffness-to-weight ratios. Similarly, L. Wang et al. (2023) proposed a multiscale numerical framework using homogenization theory to investigate the influence of geometric and coating parameters on the effective stiffness of metal-coated BCC lattice structures.

RBF have been employed in the modeling of heterogeneous lattice structures with spatially varying material properties, as demonstrated by Y. Liu, H. Yang, et al. (2021). They have also proven effective in the design of functionally graded lattice structures, where they serve as a robust interpolation scheme to map the output of TO onto spatially distributed unit cell geometries (Simsek et al. 2020).

These surrogate models are fast to evaluate and easy to differentiate, making them particularly suitable for gradient-based optimization workflows. However, their main limitation lies in their lack of uncertainty quantification, their sensitivity to the quality of the initial data, and their limited expressiveness in high-dimensional or highly nonlinear problems.

Probabilistic Models

Probabilistic surrogate models aim to not only approximate the response of a system but also provide a measure of confidence in their predictions. They model the system response as a realization of a stochastic process, most commonly a Gaussian process, which enables the estimation of both mean response and prediction uncertainty.

Among probabilistic surrogate modeling techniques, Kriging, also referred to as Gaussian Process Regression (GPR), is one of the most established and widely used methods (Kleijnen 2009). It constructs a predictive model by assuming that the unknown function to be approximated follows a Gaussian process, characterized by a mean function and a covariance kernel. The mean function represents the expected value of the response, while the covariance kernel encodes the spatial correlation between data points. The

sampling strategy plays a crucial role in this method, as it strongly influences the final accuracy of the surrogate model. Numerous sampling strategies have been developed to enhance the efficiency and robustness of Kriging models, including space-filling designs such as Latin Hypercube Sampling (LHS) or Sobol sequences. In particular, sequential sampling approaches iteratively enrich the training set by selecting new data points based on the model's predictive uncertainty, like the Expected Improvement criterion, which balances exploration and exploitation to maximize the gain of information at each step. Kriging metamodels have demonstrated strong potential in the optimization and design of lattice structures by enabling accurate predictions of mechanical responses with limited simulation data. In the work of X. Liu et al. (2022), a dual Kriging-based strategy was developed for the design of functionally graded cellular structures, where one metamodel approximates the effective elastic properties of lattice unit cells, while a second one captures the shape interpolation required for morphological smoothness between neighboring cells. This approach reduced the computational cost of multiscale optimization and ensures continuous variation in geometry, mechanical properties, and density fields. Similarly, J. Song et al. (2019) used a Kriging-assisted multi-objective GA to optimize the geometry of octet-truss lattices for maximizing stiffness and energy absorption capabilities. Co-Kriging extends standard Kriging by integrating both high-fidelity and low-fidelity data, enabling the construction of surrogate models that exploit inter-fidelity correlations to improve predictive accuracy (Dowd and Pardo-Igúzquiza 2024). By handling multi-fidelity datasets, co-Kriging can significantly reduce the number of expensive high-fidelity simulations.

More generally, Bayesian regression models form a class of methods in which uncertainty is modeled by assigning probability distributions to the model parameters or functions (Van De Schoot et al. 2021). As new data becomes available, these initial assumptions, called priors, are updated to produce a posterior distribution that reflects both the prior knowledge and the observed data. This approach makes it possible to explicitly account for model uncertainty and to incorporate prior information—such as known trends or smoothness—into the construction of the surrogate model. Bayesian regression also enables probabilistic reasoning and informed decision-making under uncertainty, which is particularly useful in applications like reliability-based design and robust optimization. The relevance of Bayesian regression models for surrogate construction has recently been demonstrated in the context of lattice structure optimization. Gongora et al. (2024) proposed a framework combining Gaussian Process regression with Bayesian optimization to design high-performance TPMS lattices. Their approach achieved high predictive accuracy on mechanical properties such as the effective Young's modulus

and enabled a 5-fold reduction in the number of simulations required compared to a classical grid-based design exploration. Similarly, Shimoyama and Komiya (2022) applied a multi-objective Bayesian optimization framework to the TO of lattice-structured heat sinks operating in natural convection. These models are especially valuable in design and optimization settings where model uncertainty can be used to guide sampling (e.g., through Bayesian optimization). Their main limitation lies in scalability, as the computational cost increases cubically with the number of training points (Stein et al. 2017), and in the difficulty of handling highly nonlinear or high-dimensional functions.

In summary, probabilistic surrogate models offer several advantages that make them particularly attractive for design and optimization tasks. They provide not only accurate predictions of the system response but also an associated measure of uncertainty. Their ability to incorporate prior knowledge and to efficiently exploit limited datasets makes them well-suited to expensive simulation contexts. However, these benefits come with limitations. First, the scalability of Gaussian process models is constrained by their $\mathcal{O}(N^3)$ computational complexity, which becomes prohibitive for large datasets (Stein et al. 2017). Second, their performance degrades in high-dimensional or highly nonlinear design spaces, where kernel selection and hyperparameter tuning become challenging.

Neural Networks-based Surrogate Models

Neural networks have gained significant popularity in recent years as powerful surrogate modeling techniques. They are particularly well-suited for high-dimensional and complex problems, where traditional surrogate models may struggle to capture intricate relationships between input and output variables. Inspired by the structure and function of biological neurons (McCulloch and Pitts 1943), neural networks consist of interconnected layers of artificial neurons, as illustrated in Figure 2.16(a). Each neuron performs a simple mathematical operation: a linear aggregation of its inputs followed by a nonlinear activation function (see Figure 2.16(b)). Mathematically, the output y of a single neuron is expressed as:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.7}$$

where x_i are the inputs, w_i the associated weights, b is the bias, and $\sigma(\cdot)$ is the activation function. In the case of the ReLU activation (Rectified Linear Unit), we have:

$$\sigma(z) = \max(0, z) \tag{2.8}$$

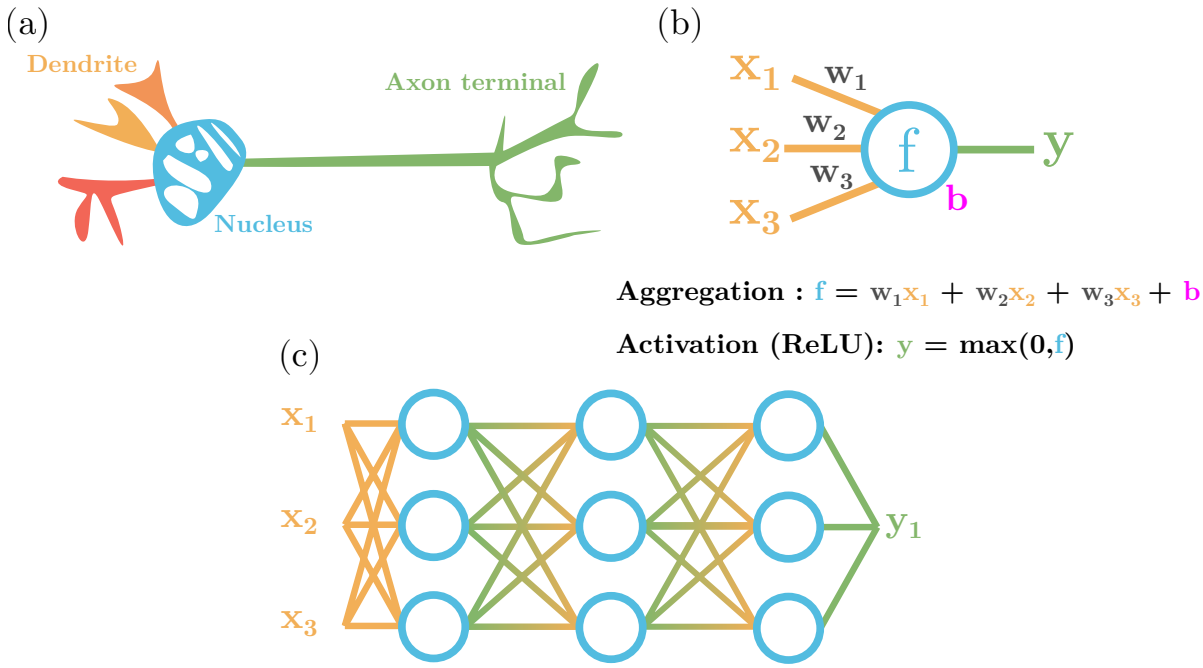

Fig. 2.16 Illustration of neural network concepts and architecture. (a) biological neuron, (b) perceptron and associated mathematical description, and (c) example of a fully connected multilayer perceptron composed of three hidden layers with three neurons per layer.

By stacking multiple neurons into layers and connecting these layers, more expressive architectures known as multilayer perceptrons are formed (Figure 2.16(c)). The network is trained on a dataset of input-output pairs by adjusting the weights and biases to minimize a loss function that quantifies the error between predicted and target outputs. Neural networks form the foundation of many modern deep learning approaches (Herrmann and Kollmannsberger 2024). To address the diversity of data types encountered, researchers have developed a wide range of specific architectures, including:

- **Convolutional Neural Network (CNN)**: Specialized for processing grid-like data, such as images, by applying convolutional filters to capture local patterns.
- **Recurrent Neural Network (RNN)**: Designed for sequential data, allowing information to persist through time steps.
- **Graph Neural Network (GNN)**: Tailored for graph-structured data, enabling the modeling of relationships between nodes and edges, which is particularly useful for molecular or material structures.

- **Autoencoder (AE)**: Used to generate a compressed transformation of input in a latent space, allowing for dimensionality reduction and feature extraction.
- **Transformers**: Based on self-attention mechanisms, these models are state-of-the-art for natural language processing and are increasingly applied in computer vision.
- **Generative Adversarial Network (GAN)**: Composed of two networks, a generator and a discriminator, that compete against each other, widely used for data generation tasks.

As the field of lattice structures continues to attract significant research interest, the application of deep learning techniques has been widely explored. CNN have been employed for various tasks. Notably, 3D-CNNs have been used to predict the homogenized mechanical properties of lattice structures (C. Liu and S. Li 2023; Rao and Y. Liu 2020). Such rapid property predictions are crucial for accelerating computationally intensive optimization frameworks, such as TO. Another notable application of CNN is found in the work of M. Zhao, X. Li, Yan, et al. (2025), who developed a model to predict the full stress–strain response of 3D lattice structures under uniaxial compression, with the aim of optimizing designs for energy absorption applications. In the context of AM, CNN-based approaches have also proven effective for quality control. For instance, Y. Zhang, Z. Zhang, et al. (2022) demonstrated the use of CNNs for detecting defects in 3D-printed lattice structures from Computed Tomography (CT) images. In the recent study by Meyer et al. (2024), a novel application of RNN was proposed for the inverse design of plate-lattice metamaterials. The authors developed an iterative modeling framework in which the lattice structure is constructed sequentially, plate by plate, enabling the prediction of geometries that correspond to target homogenized stiffness tensors.

Truss lattice structures exhibit a topology that can be naturally and compactly represented as a graph. A graph is a mathematical construct composed of nodes (or vertices) and edges, which represent connections between the nodes. GNN are a class of neural networks specifically designed to operate on such graph-structured data (Chami et al. 2022). Several studies have successfully applied GNN to predict the mechanical properties of truss lattice structures (Dold and Aranguren Van Egmond 2023; Frey et al. 2025; Maurizi et al. 2022; Zheng et al. 2023). The ability to rapidly estimate these properties enables efficient structural optimization by significantly reducing the reliance on costly numerical simulations.

More complex generative models, such as GAN, have also been employed for the creation of lattice structures with enhanced mechanical performance (Yüksel et al. 2024).

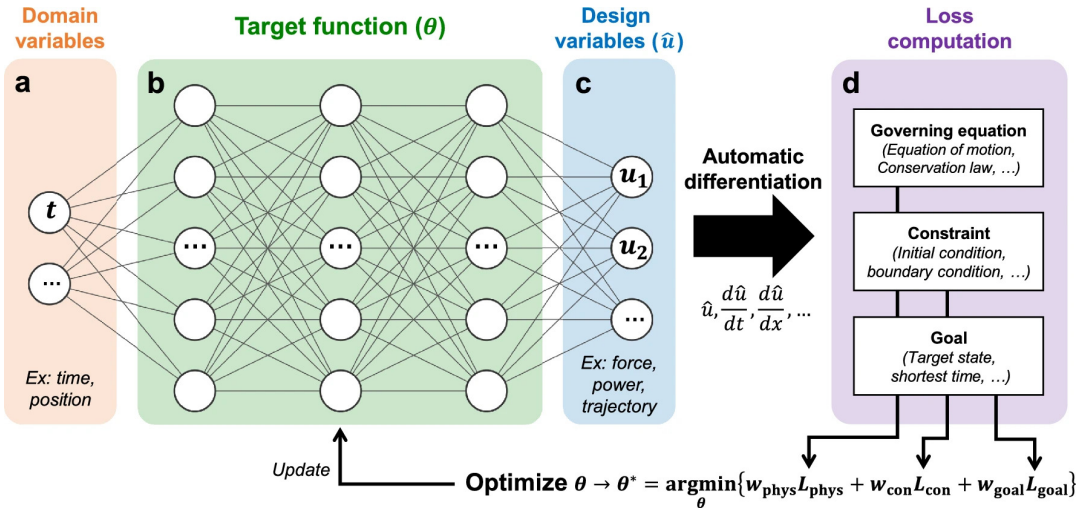

Fig. 2.17 Illustration of Physics-Informed Neural Network architecture. (a) Input variable for example position, time..., (b) multi-layer perceptrons, (c) output variable for example displacement, velocity..., and (d) the loss computation from Automatic Differentiation (AD), the loss is composed of multiple terms from physical laws, constraints, and specific objective. Reproduced with permission from (Seo 2024).

Duan and D. Wu (2025) further extended this approach by integrating both mechanical performance objectives and AM parameters within a design framework. This method allows for the simultaneous optimization of the lattice geometry and the associated printing process, ensuring that the resulting structures are not only mechanically efficient but also manufacturable.

All the architectures mentioned above rely on the concept of supervised learning, where the model is trained on a dataset composed of input–output pairs. In the context of physical systems, this typically involves using high-fidelity simulations to generate the training data. However, such approaches do not leverage the vast body of scientific knowledge already available. To address this limitation, several methods have been proposed to integrate physical knowledge directly into the training process, allowing models to benefit from both data and established physical laws.

Physics-Informed Neural Network (PINN) are Neural Network (NN) that embed physical laws, typically expressed as Partial Differential Equation (PDE), directly into the training process through the loss function. Initially introduced to address forward and inverse problems governed by PDE, PINN approximate the solution field by training a network whose outputs are constrained to satisfy the governing equations and boundary conditions. A central feature of this approach is the use of AD to compute derivatives of the network with respect to its inputs, which allows the residuals of the PDE to be evaluated efficiently (see Figure 2.17). Let us consider the one-dimensional heat equation

as a representative example:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L],\ t \in [0, T], \tag{2.9}$$

with initial and boundary conditions:

$$u(x, 0) = u_0(x), \text{(initial condition)} \tag{2.10a}$$

$$u(0, t) = u_L(t), \quad u(L, t) = u_R(t), \text{(Dirichlet boundary conditions)}. \tag{2.10b}$$

A PINN approximates the solution $u(x, t)$ by a NN $\hat{u}(x, t)$. The loss function is then constructed to penalize deviations from the PDE and its constraints:

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{IC}} + \mathcal{L}_{\text{BC}}, \tag{2.11}$$

with:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \frac{\partial \hat{u}}{\partial t}(x_i^f, t_i^f) - \alpha \frac{\partial^2 \hat{u}}{\partial x^2}(x_i^f, t_i^f) \right|^2, \tag{2.12a}$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_0} \sum_{i=1}^{N_0} \left| \hat{u}(x_i^0, 0) - u_0(x_i^0) \right|^2, \tag{2.12b}$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left(\left| \hat{u}(0, t_i^b) - u_L(t_i^b) \right|^2 + \left| \hat{u}(L, t_i^b) - u_R(t_i^b) \right|^2 \right), \tag{2.12c}$$

where $\{(x_i^f, t_i^f)\}$ are collocation points in the domain, $\{x_i^0\}$ are initial condition points, and $\{t_i^b\}$ are boundary time points. All derivatives are computed via AD. This formulation enables the NN to approximate the solution of the heat equation while satisfying physical constraints, even in the absence of labeled data throughout the domain, thereby requiring fewer training data and exhibiting better generalization capabilities under unseen conditions due to the integration of physical knowledge.

PINN have been successfully applied to a wide range of engineering problems, including fluid dynamics (Cai et al. 2021) and solid mechanics (H. Hu et al. 2024). In the context of lattice structures, the application of PINN is beginning to yield promising results. H. Wang et al. (2025) proposed a hybrid model that integrates a GNN with a physics-informed loss function to enable rapid and accurate structural analysis of hollow concrete structures featuring morphed honeycomb configurations. This design approach and NN architecture effectively reduce inconsistencies in node displacement predictions across

the entire structure, thereby facilitating the use of the surrogate model for optimization purposes.

Physics-Augmented Neural Network (PANN) extend the concept of PINN by embedding physical knowledge directly into the network architecture or training procedure. Instead of enforcing physics solely through the loss function, PANN incorporate physical invariants, symmetries, or conservation laws explicitly into the model design, such as through tailored activation functions, constrained layers, or specific network architectures. This integration enhances the model's capacity to capture complex physical behaviors while maintaining consistency with governing principles and improving interpretability. PANN are particularly advantageous for modeling advanced material behaviors, such as nonlinear elasticity or plasticity, where conventional NN may fail to respect the underlying physical structure. For example, Klein et al. (2022) and Rosenkranz et al. (2024) applied PANN to capture electromechanical and viscoelastic responses with increased physical fidelity. A noteworthy example is the NN-EUCLID framework (Thakolkaran et al. 2022), which implements a hyperelastic material model using a PANN architecture that enforces material stability and convexity, leading to improved robustness and accuracy in mechanical predictions. Furthermore, Stollberg et al. (2025) integrated a PANN trained on homogenized stiffness tensors into a multiscale TO framework for functionally graded lattice structures.

Recent advances have introduced operator learning frameworks, which aim to learn mappings between function spaces rather than finite-dimensional vectors. **Deep Operator Networks** (DeepONets) (He et al. 2023) and **Fourier Neural Operators** (FNOs) (Z. Li et al. n.d.) are examples of such approaches, well-suited for solving parametric PDEs and inverse problems.

The field of NN is rapidly evolving, with new architectures and techniques continuously emerging, enabling the modeling of increasingly complex systems. New approaches offer new perspectives for solving complex problems in physics or engineering. In the context of lattice structures, deep learning has been successfully applied to a variety of tasks, including mechanical property prediction, microstructure characterization, and acceleration of optimization processes, to name a few.

NN-based surrogate models offer several advantages for the modeling and optimization of complex physical systems. Their ability to approximate highly nonlinear relationships in high-dimensional spaces makes them particularly effective when classical surrogate models, like polynomial regression or Kriging become inefficient or intractable. Integrating physical constraints, as in PINN or PANN frameworks can improve generalization, reduce the need for large datasets, and ensure physically consistent predictions. Furthermore,

once trained,NN offer fast inference times, which is particularly valuable in iterative processes such as optimization or uncertainty quantification.

However, these approaches also come with limitations. Training deep NN requires careful hyperparameter tuning, large memory resources, and potentially long computation times, especially when the physical constraints need AD. Moreover, the black-box nature of standard NN raises concerns about interpretability and trustworthiness, as it can be challenging to understand how the model arrives at its predictions. Finally, their performance can be sensitive to the quality and diversity of the training data, and may degrade when extrapolating far from the training dataset.

2.4.2 Integration Level of Surrogate Models within the Computational Workflow

Surrogate models can be integrated at various stages of the computational workflow, either to replace expensive simulations, substitute complex constitutive models, or approximate homogenized material properties. The integration strategy depends on the targeted application and the required level of accuracy. In this context, surrogate models can be classified according to their degree of integration within the simulation or optimization pipeline, from loosely coupled to fully embedded schemes. Each level presents its own advantages and limitations in terms of computational efficiency, flexibility, and fidelity to the reference models. This section provides an overview of the different levels of integration of surrogate models and discusses their relevance for the simulation and optimization of lattice structures. A summary of the different integration levels is presented in Figure 2.18.

Global Response Prediction

One common strategy for integrating surrogate models operates at the global response level, aiming to predict effective mechanical properties from full-scale simulations. Such simulations are often computationally expensive, particularly when dealing with large-scale problems or complex geometries such as lattice structures. The principle of this approach is to run a limited number of high-fidelity simulations and then train a surrogate model to interpolate or extrapolate the mechanical response for unseen parameter configurations. For instance, M. Zhao, X. Li, Yan, et al. (2025) proposed a CNN-based surrogate model to predict the effective stress–strain response of 3D lattice structures under uniaxial compression. Similarly, Donnici et al. (2024) focused on predicting the three-point bending stiffness of beams filled with Octet and IsoTruss lattice

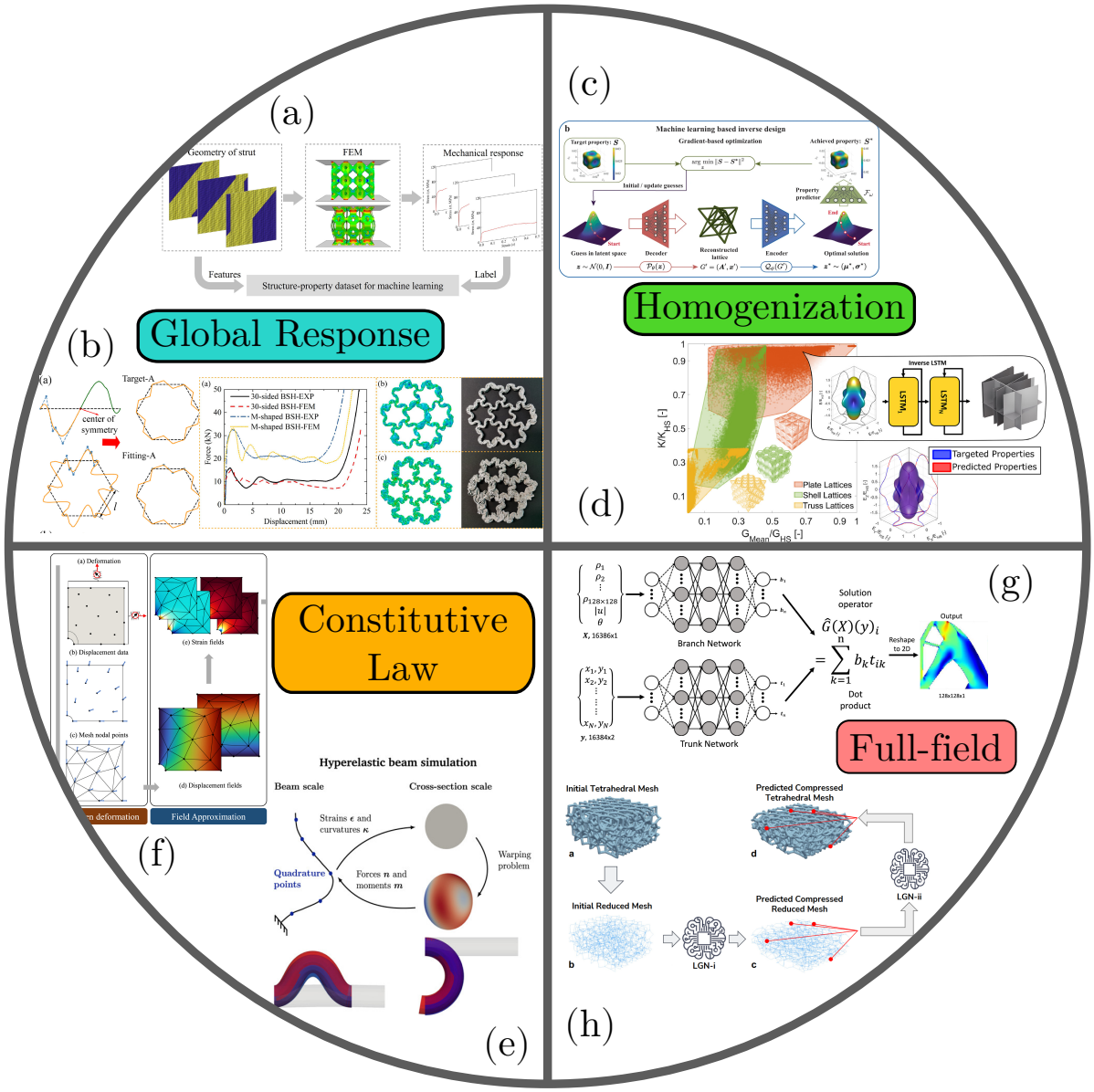

Fig. 2.18 Overview of the multiple integration level of surrogate models. Adapted with permission from (a) (M. Zhao, X. Li, Yan, et al. 2025) (b) (Zhou et al. 2025) (c) (Zheng et al. 2023) (d) (Meyer et al. 2024) (e) (Schommartz et al. 2024) (f) (Thakolkaran et al. 2022) (g) (He et al. 2023) (h) (Jain et al. 2024).

topologies. Another applied example, Alkentar and Mankovits (2025) carried out full-scale simulations of hip implants incorporating lattice infill and trained a surrogate model to predict the effective mechanical response of these devices. In other domains, such as thermal physics, surrogate models have also been employed to reduce the computational cost of Computational Fluid Dynamics (CFD) simulations for lattice structures, notably to optimize the topology of heat sinks (Shimoyama and Komiya 2022). This approach is particularly effective for accelerating the design process, as it enables rapid exploration of high-dimensional design spaces. It can be combined with metaheuristic optimization algorithms to efficiently guide the search for optimal configurations. For instance, Zhou et al. (2025) proposed a surrogate-assisted evolutionary algorithm to optimize the geometry of lattice structures for energy absorption applications. However, a major limitation is the restricted generalization ability of the surrogate model, which is inherently tied to the diversity of the training data. For instance, a model trained solely on compression load cases will generally fail to accurately predict the structural response under other loading conditions, even if they are relatively similar. The primary reason for limiting the number of geometrical parameters lies in the high computational cost associated with generating the training dataset, as well as the increased complexity required of the surrogate model to accurately capture the underlying relationships.

Homogenized Property Prediction

The use of simplification techniques such as homogenization is a widely adopted strategy to reduce the computational cost of simulating complex structures, as discussed in Section 2.2.2. Even at the unit cell scale, predicting the complete mechanical behavior of lattice structures can remain computationally demanding. Homogenization provides an effective way to reduce the dimensionality of the output space, enabling the use of surrogate models to approximate effective mechanical properties rather than full-field responses. A wide range of surrogate modeling techniques have been employed at this level, with varying levels of complexity in the design space. Examples include Kriging-based models applied to functionally graded TPMS lattices (X. Liu et al. 2022), as well as RNN used for the iterative design of non-symmetric plate lattice structures (Meyer et al. 2024). A particularly notable example involving high-dimensional truss design is the work of Zheng et al. (2023), who proposed a generative model capable of spanning a wide range of property values and demonstrated good extrapolation capabilities beyond the training data. It has also been shown that this approach can be extended to predict multiple physical properties. For instance, Frey et al. (2025) used a GNN to simultaneously predict the effective thermal and mechanical properties of multi-material

truss lattices in the context of multi-objective optimization. In definitive, the objective of this homogenization-based approach is to identify cell geometries that exhibit specific effective properties tailored to a given application. It has been extensively used in the context of TO to optimize the design of lattice structures (Jansari et al. 2025; J. Wang and Panesar 2022; Woldseth et al. 2022). While this method is highly attractive for the design of lattice structures, it is not without limitations. As previously mentioned, the underlying assumption of scale separation inherent to homogenization restricts its applicability to cases where the global structural response is decoupled from local phenomena. This constraint may limit its effectiveness in cases involving high variation gradients or complex boundary conditions.

Constitutive Law Substitution

At the most fundamental level, surrogate models can be employed to directly replace the material constitutive laws within a FE formulation. This strategy is particularly valuable when dealing with nonlinear, history-dependent, or anisotropic material behaviors that are difficult to capture with classical models. Recent advances in this direction include the use of NN-based models trained to emulate the stress–strain relationship of complex materials, either in a supervised or unsupervised setting. For instance, Klein et al. (2022) proposed a PANN approach to model finite electro-elasticity while enforcing key physical constraints such as objectivity and thermodynamic consistency. In a different context, Thakolkaran et al. (2022) introduced the NN-EUCLID framework, which enables unsupervised learning of hyperelastic constitutive models directly from full-field displacement and global reaction force data, bypassing the need for stress data. More recently, Schommartz et al. (2024) used the concept of PANN to the modeling of geometrically exact hyperelastic beams. Their approach constructs a neural representation of the beam strain energy potential as a function of axial strain and curvature, from which internal forces and moments are derived via differentiation, ensuring thermodynamic consistency. The model accounts for geometric and material nonlinearities, includes physical constraints such as objectivity, normalization, and symmetry, and can be parameterized with geometric quantities such as beam radius.

These approaches open the door to fully data-driven simulations where the material response is embedded in a differentiable model, allowing for direct integration within optimization loops and efficient evaluation across complex design spaces. However, the training of such models often requires high-fidelity data generated either through full-field simulations or advanced experimental techniques, which can be computationally and experimentally intensive.

Full-field Response Prediction

A complex but promising strategy consists in directly predicting the mechanical response of the fully meshed structure using a surrogate model. This approach enables the simulation to be carried out solely through the surrogate, significantly accelerating the computational process while preserving detailed information on deformation and stress at every point within the structure. The main advantage lies in its ability to predict structural behavior at a very low computational cost. However, a major challenge remains in training a model capable of accurately generalizing the physical response across the entire structure domain. He et al. (2023) successfully trained a DeepONet architecture to predict stress fields in nonlinear elasto-plastic materials. Their training relied on results obtained from TO simulations under various loading conditions. Similarly, the work of Jain et al. (2024) proposed a GNN-based approach to simulate high-fidelity responses of 3D truss lattice structures made of hyperelastic materials. Their model, LatticeGraphNet, achieves an average pointwise prediction error below 1.67% relative to the structural dimensions. With a similar GNN approach, Maurizi et al. (2022) developed a model to predict the stress–strain response in various material systems, like fiber, stratified composites, and truss lattices. While this approach shows strong potential for accelerating mechanical simulations, it requires an extremely costly training phase, and its predictive accuracy remains strongly dependent on the structure size. Furthermore, generalization to geometries or loading conditions outside the training distribution remains a critical limitation.

Concluding Remarks on Surrogate Model Integration Levels

In summary, integrating surrogate models at various levels within the computational workflow offers distinct trade-offs in terms of accuracy, computational cost, and flexibility. From predicting global responses to replacing full-field simulations or constitutive laws, each strategy serves a specific purpose depending on the complexity of the structure and the objectives of the analysis or optimization task. In the context of lattice structures, even when limited to truss-based designs, exploring the design space remains computationally demanding. Selecting the most appropriate level of surrogate model integration is therefore a non-trivial task. It requires careful consideration of multiple factors, including the level of detail required, the physical accuracy needed, the availability of high-fidelity data, and the compatibility with the optimization framework. Ultimately, the choice of surrogate model and its integration level should be driven by the targeted application and the balance between predictive performance and computational efficiency.

2.4.3 Potential and Limitations of Surrogate Modeling

As discussed in this section, surrogate models have emerged as powerful tools for accelerating the design and optimization of complex structures, particularly in the context of lattice architectures. By reducing the computational burden associated with high-fidelity simulations, surrogate models enable the exploration of larger design spaces and the identification of improved solutions within reasonable time frames. In the context of lattice structure optimization, surrogate models offer several notable advantages:

- They significantly reduce the number of high-fidelity simulations required to explore the design space.
- Most surrogate models are differentiable, enabling the use of gradient-based optimization techniques.
- Their low computational cost allows for the evaluation of larger design space.

Nevertheless, several limitations must be carefully considered:

- The method needs an accurate dataset and careful model selection.
- The predictive accuracy of a surrogate model strongly depends on the quality, coverage, and diversity of the training dataset.
- Surrogate models, particularly those based on deep NN, often suffer from limited interpretability.
- In high-dimensional design spaces, maintaining accuracy becomes challenging without appropriate data selection, feature engineering, or dimensionality reduction.
- Generalization to unseen configurations remains a major issue, especially for extrapolation outside the training distribution.

Ultimately, the selection of an appropriate surrogate model and its level of integration within the computational workflow should be guided by the specific application, the required accuracy, and the acceptable computational cost. Although surrogate modeling is a promising strategy to accelerate the design process, it requires a careful balance between efficiency and physical fidelity. A key risk of this approach is the potential loss of physical consistency and the difficulty in clearly defining the model's domain of validity. However, constructing advanced surrogate models that embed physical laws into their architecture represents a promising strategy to overcome these limitations, as demonstrated by PANNs.

2.5 Scientific Challenges and Positioning

The complexity enabled by new AM processes is opening novel design opportunities for more intricate and efficient structures. Lattice structures represent a promising strategy to explore this design space, as their periodic nature simplifies the geometrical parametrization, property evaluation through multiscale methods, and printability. These new architectures increase the diversity of attainable physical properties, thereby unlocking novel application domains. However, this rise in complexity brings significant challenges throughout the entire digital chain, from CAD modeling to manufacturing. A global and unified vision of every stage in the design process is required to fully exploit the potential of these structures. Segmenting the process into isolated steps may diminish the coherence and relevance of the overall methodology. The first challenge is to design lattice structures with well-defined and controllable parametrization, allowing a thorough exploration of the design space. In this thesis, we focus on truss lattice structures for several reasons. First, these structures demonstrate excellent mechanical performance, particularly in terms of energy absorption and stiffness-to-weight ratio. Second, their attainable complexity with a limited number of parameters is higher compared to other structural types. Third, their mechanical response can be evaluated efficiently using beam theory.

However, a review of the literature in section 2.2.1, reveals that the assumptions underlying classical beam theory significantly restrict the accurate prediction of the mechanical properties of truss lattice structures. While some studies have improved the applicability of beam theory by introducing corrections at the joints, yet no generalized approach is proposed for all truss lattice structures. Our first contribution, presented in Chapter 3, is the development of an optimal penalty method to correct the joint stiffness in truss lattice structures. The proposed method employs a geometrical function to define the joint region between beam elements, enabling the correction of the appropriate amount of material for every type of lattice geometry.

After establishing a reliable and efficient method to predict the mechanical properties of truss lattice structures, the next challenge is to design these structures with specific mechanical properties. Cell geometry optimization is extensively studied in the literature, with the objective of finding specific homogeneous properties or global responses, like negative Poisson's ratio. These studies are limited by using a single optimized cell geometry in the entire structure. Finding the optimal arrangement of multiple cell geometries in a lattice structure is a more complex problem, because of the number of involved parameters. However, this approach holds considerable potential, as it enables the design of structures with locally tailored mechanical properties throughout the entire

architecture. The major challenge in this approach is the computational cost of evaluating the mechanical properties of the entire lattice structure with one or multiple parameters per cell. The common approach is to use homogenization to reduce the computational cost. However, the homogenization method is limited to periodic structures, and the assumption of scale separation may not be valid. In this thesis, we propose a new approach for simulating lattice structures with multiple cell geometries, based on a domain decomposition method combined with a surrogate model, specifically a PANN, in order to predict the mechanical properties of the entire structure within reasonable computational time. Additional advantages of this methodology will be discussed in Chapter 4. Finally, the last challenge addressed in this thesis concerns the integration of the simulation method into an optimization framework. The objective is to determine the optimal distribution of cell geometries within a lattice structure in order to achieve targeted mechanical properties, such as maximizing stiffness and minimizing weight. This optimization problem is particularly demanding due to the high dimensionality of the design space and the computational expense associated with evaluating the mechanical response of the entire structure. To overcome these difficulties, we introduce a gradient-based optimization framework that exploits the differentiability of the surrogate model presented in Chapter 4, and is further developed in the context of lattice optimization in Chapter 5.

Chapter 3

Enhanced Beam Modeling for Lattice Structures: A Generalized Joint Stiffening Approach

"A journey of a thousand miles begins with a single step." — Lao Tzu

Contents

3.1 Beam Models for Lattice Structures: An Efficient Yet Challenging Approach 62
3.1.1 Beam Model Formulation 62
3.1.2 Challenges in Modeling Joint Stiffness in Beam Models 64
3.1.3 Lattice Geometries, Material Properties, and Experimental Framework 67
3.2 Identification of Optimal Stiffening Parameters via Static Analyses 70
3.2.1 Assessment of the Beam Models Accuracy over Unit Cells 70
3.2.2 Defining the Length of the Stiffening Region 72
3.2.3 Defining the Penalty Coefficient 74
3.2.4 Evaluating the Effects of the Stiffening Region on Beam Model Accuracy 74
3.2.5 Optimizing the Penalty Coefficient 76
3.3 Application of the Joint Stiffening Method to Dynamic Simulations 77
3.3.1 Assessment of the Beam Models Accuracy via Modal Analyses . . . 78
3.3.2 Balance the Mass Distribution 80
3.3.3 Evaluating Added Mass Effects on the Structural Dynamics 80
3.4 Extension to Nonlinear Analyses and Experimental Comparison 82
3.4.1 Uniaxial Compression of a Unit-cell Made of Elasto-Plastic Material 82
3.4.2 Experimental Validation on 3D-printed BCC Structures 84
3.5 Conclusion on the Beam Model Penalization Methodology . . . 87

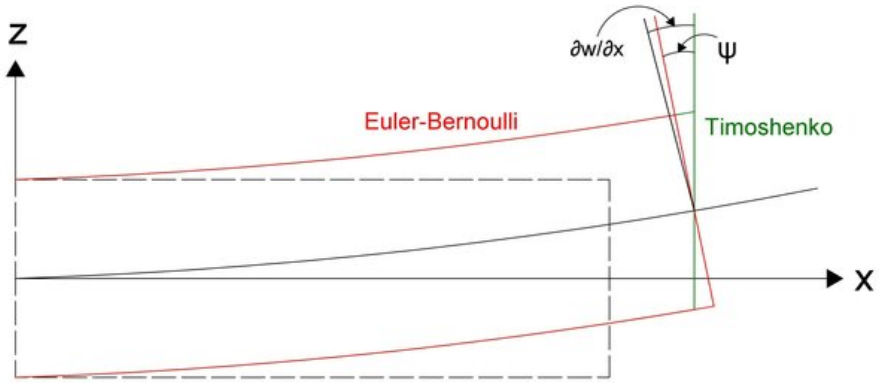

Fig. 3.1 Comparison of the deformations of a beam according to the Timoshenko and Euler-Bernoulli beam theories. The angle Ψ represents the rotation of the cross-section in the Timoshenko theory, while $\frac{\partial \omega}{\partial x}$ denotes the slope of the deflected neutral axis. Reproduced with permission from (Gulkanov and Modestov 2019).

3.1 Beam Models for Lattice Structures: An Efficient Yet Challenging Approach

Truss-based lattice structures are widely used due to their attractive mechanical properties, such as controllable anisotropy and high SEA. In addition, their behavior modeling remains relatively straightforward, as the structure can typically be represented as a network of beams with constant cross-section. However, achieving an accurate simulation of these structures requires a fine discretization, which becomes prohibitive both in terms of mesh file size and computational time for large-scale structures. To overcome this issue, beam models are often employed as a simplified yet efficient alternative to full 3D solid models in the simulation of lattice structures, as discussed in Section 2.2.1.

3.1.1 Beam Model Formulation

This study focuses on a shear-deformable, Timoshenko-like 3D beam model. In the present work, only straight beams are considered. While the proposed framework can be naturally extended to curved beam geometries, this simplified configuration constitutes a meaningful first step toward more general and complex beam models. Let us recall the basics of these models in the context of linear elasticity. The initial configuration of the beam is represented by an arc-length parameterized centerline curve $\boldsymbol{x}_c : [0, l] \to \mathbb{R}^3$ and

an orthonormal frame field $\boldsymbol{R} : [0, l] \to \mathrm{SO}(3)$ describing the cross-section orientation. The map from the initial configuration to the deformed configuration is described by the displacement field of the centerline $\boldsymbol{u}_\mathrm{c} : [0, l] \to \mathbb{R}^3$ and the cross-section rotational field $\boldsymbol{\theta} : [0, l] \to \mathbb{R}^3$. More specifically, the strain measures, according to the Timoshenko beam theory, can be expressed as follows:

$$\boldsymbol{\epsilon} = \begin{pmatrix} e_t \\ \gamma_1 \\ \gamma_2 \end{pmatrix} = \boldsymbol{R}^\top \boldsymbol{u}_\mathrm{c}' + \boldsymbol{e}_1 \times \boldsymbol{\theta}, \qquad \boldsymbol{\kappa} = \begin{pmatrix} \kappa_t \\ \kappa_1 \\ \kappa_2 \end{pmatrix} = \boldsymbol{\theta}', \tag{3.1}$$

where $(\cdot)'$ denotes the arc-length derivative, $\boldsymbol{e}_1$ the first Cartesian vector, $\boldsymbol{\epsilon}$ the tensile and shear strain vector, and $\boldsymbol{\kappa}$ the torsional and bending strain vector. For an isotropic elastic material, the constitutive relation can be characterized by a Young's modulus E and a shear modulus G. The internal forces $\boldsymbol{n}$ and moments $\boldsymbol{m}$ are then given by:

$$\boldsymbol{n} = \boldsymbol{A}\boldsymbol{\epsilon}, \quad \boldsymbol{A} = \mathrm{diag}\begin{pmatrix} EA & GA_1 & GA_2 \end{pmatrix}, \tag{3.2}$$

$$\boldsymbol{m} = \boldsymbol{B}\boldsymbol{\kappa}, \quad \boldsymbol{B} = \mathrm{diag}\begin{pmatrix} GJ & EI_1 & EI_2 \end{pmatrix}, \tag{3.3}$$

where A, A_1, and A_2 defines the cross-section area and shear areas, respectively, and J, I_1, and I_2 defines the torsional constant and second moments of area, respectively. Shear and torsion are often scaled with correction factors, *i.e.*, in general $A_i \neq A$ and $J \neq I_1 + I_2$. Readers may refer to (Pilkey 2002) for more details regarding Timoshenko-like beam models.

For slender beams where shear deformation can be neglected, the Euler-Bernoulli beam theory is usually utilized instead (Bauchau and Craig 2009). Here, the shear strains γ_1 and γ_2 are assumed to be zero, simplifying the formulation but limiting its validity to thin structures with high aspect ratios. Figure 3.1 illustrates the differences in terms of deformation between the Timoshenko and Euler-Bernoulli beam theories. Finally, we recall the variational formulations associated with such a beam model. The static equilibrium of a structure modeled with Timoshenko beams may involve weak problems of the following form:

$$\int_{[0,l]} \boldsymbol{n} \cdot \delta\boldsymbol{\epsilon} + \boldsymbol{m} \cdot \delta\boldsymbol{\kappa} \ \mathrm{dl} = \int_{[0,l]} \boldsymbol{b}_\mathrm{c} \cdot \delta\boldsymbol{u} \ \mathrm{dl} \ + \mathrm{BCs}, \tag{3.4}$$

where $\boldsymbol{b}_c$ denotes line loads applied along the beam centerline, and BCs denotes additional terms associated with the imposed boundary conditions (e.g., imposed displacement, imposed rotation, applied force, or applied moment).

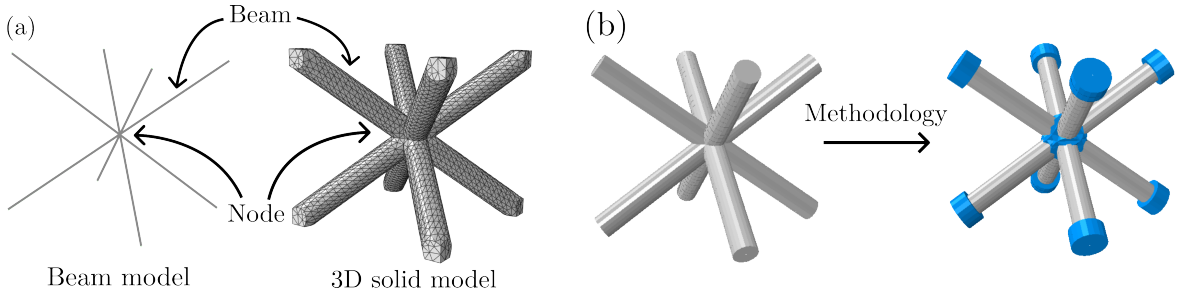

Fig. 3.2 (a) Comparison of node dimensions according to the beam model and the 3D model for the BCC (b) Methodology applied to the BCC unit cell.

3.1.2 Challenges in Modeling Joint Stiffness in Beam Models

The main limitation of beam models lies in their inability to effectively and accurately capture the stiffness of the structure at the beam junctions, particularly in the nodal regions. Figure 3.2(a) illustrates the difference in dimensions between the beam model and the 3D model for a BCC unit cell. This discrepancy can lead to an underestimation of the stiffness of the overall structure, as highlighted in several studies (Gümrük and Mines 2013; H. Guo et al. 2020; Labeas and Sunaric 2010; Luxner et al. 2005; Smith et al. 2013; Tahmasebimoradi et al. 2021).

To address this limitation, several strategies have been proposed to modify the stiffness at the junctions. Two main penalty parameters are commonly used: the length of the nodal zone denoted in this work l_{zone} , and a second parameter that increases the stiffness within this nodal zone (see Figure 3.2(b)). This stiffness increase can either be done by enlarging the cross-section via a penalty coefficient denoted here α_{r} (*e.g.*, in case of circular cross-section defining $R_{zone} = \alpha_{\text{r}}\, R$), or by increasing the Young's modulus via a penalty parameter denoted here α_{E} (*i.e.*, defining $E_{zone} = \alpha_{\text{E}}\, E$).

However, it appears that there is no consensus on the most appropriate penalty parameters for beam models of lattice structures. The choice of penalty parameters is often based on empirical observations or specific applications, leading to a wide range of values in the literature. Table 3.1 summarizes the studies that proposed penalty methods to stiffen the junctions in beam models for lattice structures, along with the corresponding key parameters. Several studies used the penalty coefficient α_{E} to increase the Young's modulus in the nodal zone. For example, Luxner et al. (2005) obtained appropriate results using $\alpha_{\text{E}} = 1000$ for joints with at least four connecting beams. The idea behind this approach is to achieve stiff nodes that behave like solid elements. The same approach was used by A. Alaimo et al. (2023) for a BCC lattice structure, but they optimized the length parameter l_{zone} using a multi-objective GA. While Gümrük and Mines (2013)

Table 3.1 Overview of simulation strategies for lattice structures reported in the literature. The main parameters involved in the joint stiffening approaches are the beam radius R, the beam length L, and the angle γ between adjacent beams at the node. The studied lattice geometries include: Simple Cubic (SC) / Gibson-Ashby Cell (GAC) / BCC / Reinforced Body-Centered Cubic (RBCC) / Body-Centered Cubic with beam on Z-axis (BCCZ) / Face Centered Cubic with beam on Z-axis (F2FCCZ) / Reinforced Body-Centered Cubic with beam in horizontal plane at center node (BCC+C) / Octet Truss (Octet) / Truncated-cuboctahedron (TCO) / Auxetic (AUX).

Reference	Modified parameter	Modification length l_{zone}	Studied lattice	Relative density	Comment
Luxner et al. (2005)	$\alpha_{\text{E}} = 1000$	R	SC GAC BCC RBCC	$[0.1, 0.15]$	Modification applied only to nodes connected to four or more beams.
Labeas and Sunaric (2010)	$\alpha_{\text{r}} = 1.4$	$L/10$	BCC BCCZ F2FCCZ	$[0.08, 0.12]$	Plasticity and buckling
Gümrük and Mines (2013)	$\alpha_{\text{E}} = 1.5$	R	BCC	$[0.04, 0.15]$	Micro lattice
Smith et al. (2013)	$\alpha_{\text{r}} = 1.4$	$L/10$	BCC BCCZ	$[0.03, 0.15]$	Plasticity and buckling
Terriault and Brailovski (2018)	$\alpha_{\text{E}} = \frac{E_{Solid}}{E_{Beam}}$	L	Tetrahedral	$[0.05, 0.95]$	Lattice from tetrahedral meshed geometry
Weeger, Boddeti, et al. (2019)	$\alpha_{\text{r}} = f(R, L, \gamma)$	$f(R, L, \gamma)$	BCC SC Octet Vintiles Cross	Not specified	Spline-based soft lattice structures
H. Guo et al. (2020)	$\alpha_{\text{r}} = 1.4$	Node cutout extraction	BCC	$[0.05, 0.2]$	Study of high-density geometries and plasticity
Tahmasebimoradi et al. (2021)	$\alpha_{\text{r}} = 1$ (hybrid method)	R	BCC BCCZ	0.05	Solid modeling of the junction including defects
A. Alaimo et al. (2023)	$\alpha_{\text{E}} = 10000$	GA	BCC BCC+C	$[0.01, 0.5]$	
Schwahofer et al. (2023)	$\alpha_{\text{E}}(\gamma) = [1, 12]$	$\frac{R}{\sin(\gamma)}$	No particular	No limit	
McDonnell et al. (2024)	$\alpha_{\text{r}} = 1.4$	R	BCC	$[0.099, 0.16]$	Plasticity and defects
De Biasi et al. (2025)	Optimize α_{r}	Dependent of geometry	BCC BCCZ TCO AUX	$[0.5, 0.1]$	Porosity compensation

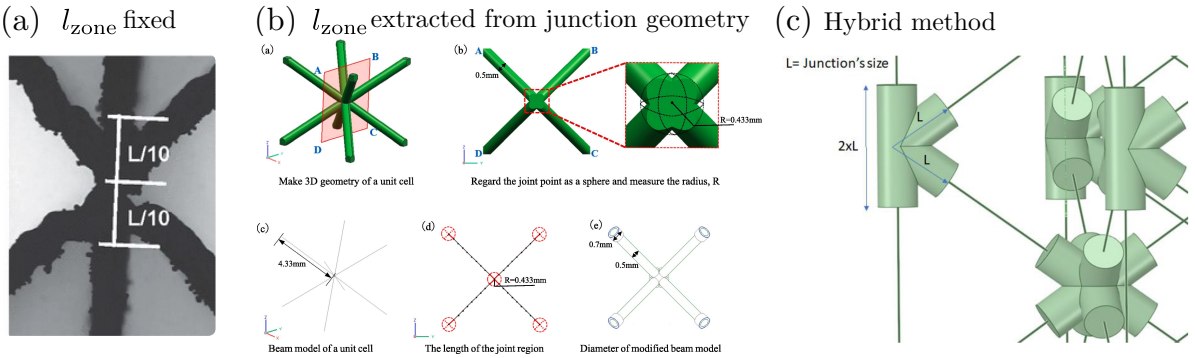

Fig. 3.3 Methodologies employed in the literature to adjust nodal stiffness in beam models. Reproduced with permission from (a) l_{zone} fixed (Labeas and Sunaric 2010), (b) l_{zone} extracted from junction geometry (H. Guo et al. 2020), (c) Hybrid method (Tahmasebimoradi et al. 2021).

recommended $\alpha_E = 1.5$ and $l_{zone} = R$ for BCC microstructures. Schwahofer et al. (2023) proposed a method to evaluate α_E based on the geometric arrangement of the node, and established a relationship between the penalty coefficient and the angle γ between the beams. To derive this relationship, they performed simulations using both beam and solid elements, and identified the value of α_E that minimizes the discrepancy between the two models.

Regarding the studies that penalized cross-section dimensions, it appears that setting the penalty coefficient as $\alpha_r = 1.4$ is frequently adopted, see for instance (H. Guo et al. 2020; Labeas and Sunaric 2010; Lozanovski, Downing, Tran, et al. 2020; McDonnell et al. 2024; Smith et al. 2013; Tahmasebimoradi et al. 2021). However, the penalty region l_{zone} varies across these studies. For example, Smith et al. (2013) and Labeas and Sunaric (2010) selected one-tenth of the strut length for lattice geometries such as BCC, BCCZ, or F2FCCZ (see Figure 3.3(a)). In contrast, H. Guo et al. (2020) employed a more general approach suitable for denser lattice structures. They defined l_{zone} by trimming a 3D geometric model at the node and determining the radius of the minimal sphere enclosing all the material at the joint (see Figure 3.3(b)). Additionally, let us mention L. Meng et al. (2020) who employed tapered beams to consider both material accumulation at the truss junctions and other geometric defects coming from AM. Weeger, Boddeti, et al. (2019) utilized a more sophisticated method: defining the radius by a quadratic B-Spline curve which leads to smooth and regular cross-section variations. Finally, an interesting approach was introduced by Tahmasebimoradi et al. (2021), who proposed a hybrid method combining solid elements in joint regions with beam elements elsewhere (see Figure 3.3(c)). This combination enables an accurate representation of the local joint behavior while preserving the computational efficiency of beam models for the

remaining structure. However, this approach requires a more complex implementation and increased computational resources.

While these methods focus on specific lattice geometries within narrow relative density ρ_{rel} (*i.e.*, the ratio between the amount of material and the volume enclosing the unit cell) ranges, several studies introduced advanced models which incorporate material or geometric nonlinearities (Gärtner et al. 2021; H. Guo et al. 2020; Lozanovski, Downing, Tino, et al. 2020; Lozanovski, Downing, Tran, et al. 2020; L. Meng et al. 2020; Smith et al. 2013; Tahmasebimoradi et al. 2021; Weeger, Valizadeh, et al. 2023). This diversity complicates the reproducibility and comparability of the results, preventing the identification of a generic strategy for a wide range of lattice structures achievable through AM.

To address this issue, we proposed a generic method to determine optimal penalty parameters l_{zone} and α_{r} for beam models of lattice structures. We developed a generalized approach that can be applied to any lattice geometries and any beam radii R. The following sections detail the strategy implemented to identify these optimal parameters in the context of static simulations. The first step consists of formulating simple static problems, combined with straightforward metrics, to evaluate the accuracy of beam models of lattice structures. Based on these metrics, the objective is to identify the optimal penalty parameters for joint stiffening. The determination of these parameters is carried out by simulating various lattice structures across a range of relative densities. In a second step, we extend the methodology to dynamic simulations, where an additional parameter is introduced to accurately capture the dynamic response of the beam model. Finally, once these penalty parameters have been determined, we investigate the relevance of the proposed approach in more advanced cases, including elasto-plastic material behavior, geometric nonlinearity, and comparisons with experimental data. Prior to the parametric analyses, we present the lattice geometries, material properties, and experimental setup used throughout this work.

3.1.3 Lattice Geometries, Material Properties, and Experimental Framework

Several lattice structure geometries have been chosen, as depicted in Figure 3.4, to explore the validity domains of the penalization strategies. Each unit cell has been selected based on its frequent appearance in the literature, such as the Body-Centered Cubic (BCC) cell. The group of unit cells is designed to include structures with a variable number of beams and angles between these beams, aiming to achieve greater diversity within the dataset.

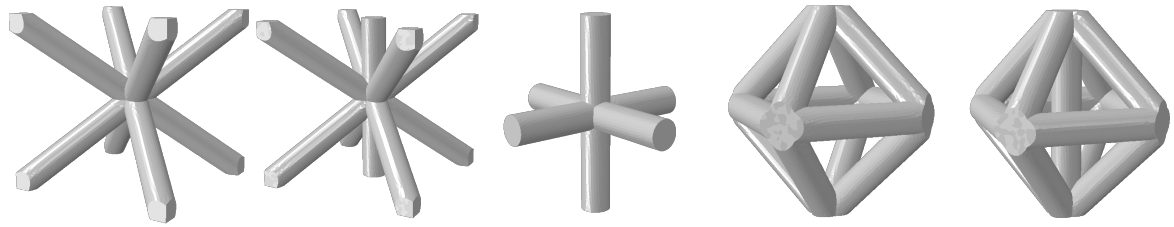

Fig. 3.4 Set of 3D solid lattice unit cells that are studied in this work.

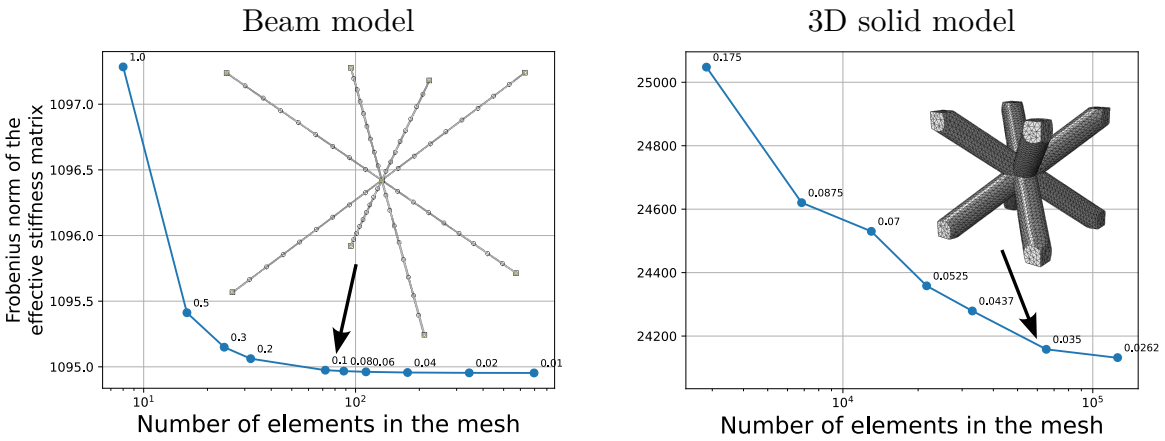

Fig. 3.5 Mesh convergence analysis for beam and 3D solid models.

All structures were studied in the radius range between 0.025 mm and 0.2 mm, with a step of 0.025 mm. By acting on the radius, the domain of relative density will depend on the structure. This variation is influenced by structural features such as the number of beams and nodes, which determine the material distribution in the cell (Z. Chen et al. 2018). Beam theory requires a high aspect ratio (beam length/beam radius) to accurately represent mechanical behavior. Thus, the domain is narrowed by calculating the ratio between the smallest beam length in the structure and its radius. For Timoshenko beam model, the limitation of the ratio is set to values greater than 4 (which already exceeds standard recommendations for these beam elements (Schwahofer et al. 2023)).

All simulations in this chapter were conducted with the commercial software Abaqus (Dassault Systèmes, Vélizy-Villacoublay, France, Version 2022) coupled with a Python script allowing rapid modeling, simulation, and extraction of results[1]. Subsequently, after performing a mesh convergence analysis shown in Figure 3.5, ten-node tetrahedral

[1]The full code used for generating the results and figures is available at https://github.com/Tcadart/Code_Beam_Penalization_methodology

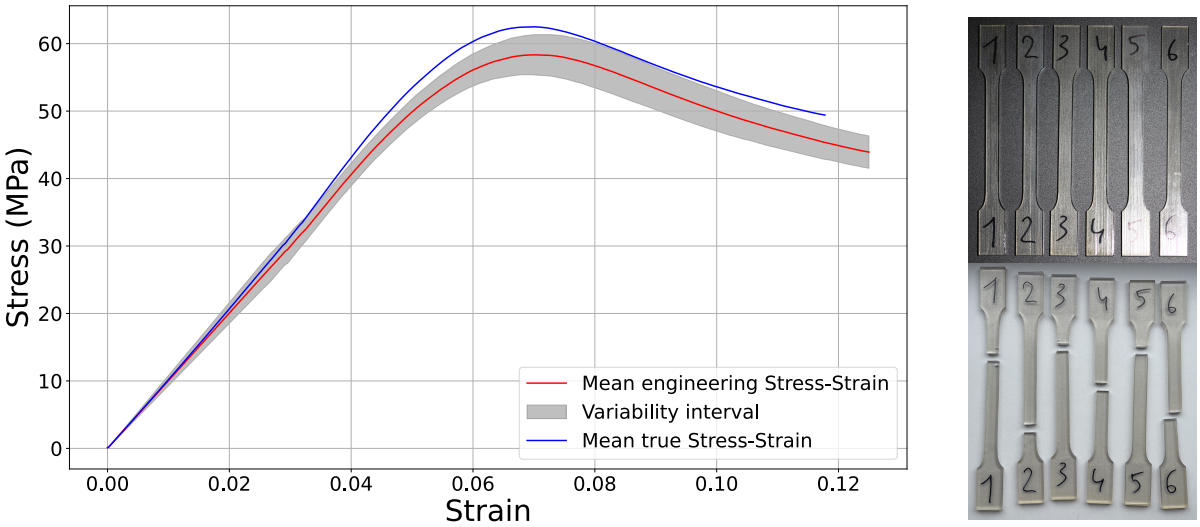

Fig. 3.6 Stress-strain response of VeroClear™ material fabricated using PolyJet 3D printing.

elements with an approximate global size of 0.0375 were selected for the 3D solid model. For the beam configuration, Timoshenko beam linear elements sized at 0.1 were adopted.

In this thesis, material jetting was selected as the manufacturing technique for all experimental tests. This AM technique offers significant advantages in terms of design flexibility and resolution. Its ability to handle multi-material deposition and to accurately reproduce fine features makes it well suited for prototyping functional lattice structures. Although parts produced by material jetting tend to be more brittle and exhibit lower mechanical strength, thermal resistance, and long-term stability compared to other AM technologies (Gülcan et al. 2021), these limitations are not critical for the purposes of this work. The objective is not to assess the ultimate mechanical performance of the printed parts, but rather to experimentally validate the numerical predictions and optimization methodology developed throughout the thesis. Consequently, the advantages of material jetting in terms of ease of use, geometric fidelity, and process reliability outweigh its drawbacks for the intended applications. The material used for both numerical and experimental investigations is VeroClear™. The constitutive model applied in the simulations is first assumed to be purely linear elastic for the initial evaluation of the penalization coefficients. The Young's modulus was experimentally identified as $E = 1013$ MPa, as illustrated in Figure 3.6. The material properties were complemented by a density of 1.18 g/cm^3 and a Poisson's ratio of $\nu = 0.3$, consistent with the data reported by Stratasys (*VeroClear* 2024). For the experimental characterization, specimens were manufactured using a PolyJet Objet260 Connex3 printer (Stratasys®, Ltd, USA). Quasi-static tensile tests were conducted on six standardized samples following

the ISO 527-2 protocol, using a Lloyd LR 50K traction-compression testing machine (AMETEK, Chicago, USA). The resulting stress-strain response of the material is reported in Figure 3.6, and was directly integrated into the simulation software as a tabular material model for subsequent nonlinear analyses.

3.2 Identification of Optimal Stiffening Parameters via Static Analyses

3.2.1 Assessment of the Beam Models Accuracy over Representative Unit Cells

We seek a straightforward method to assess beam models accuracy in the context of lattice structures. The developed approach uses common practices in the context of multi-scale homogenization. The idea is to formulate problems within representative unit cells and subsequently compute several quantities of interest in order to evaluate the accuracy of the model later on. The results obtained with the beam models are compared with those obtained with a solid model. The 3D solid model is considered as the reference, although it remains an idealized numerical approximation that may still differ from experimental observations.

More specifically, let us define a macro-displacement field $\boldsymbol{u}_0$ over an elementary volume Ω_0 by:

$$\boldsymbol{u}_0(\boldsymbol{x}) = \nabla\hat{\boldsymbol{u}}_0 \cdot \boldsymbol{x}, \quad \text{for } \boldsymbol{x} \in \Omega_0, \tag{3.5}$$

where the macro-displacement gradient $\nabla\hat{\boldsymbol{u}}_0$ is taken constant over the macro-domain. We are then looking for the fluctuation vector fields, solutions of:

$$\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\,\mathrm{dv} = -\int_\Omega (\boldsymbol{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}_0)) : \delta\boldsymbol{\varepsilon}\,\mathrm{dv} \; + \text{PBCs}, \qquad \text{(solid formulation)} \tag{3.6a}$$

$$\int_{[0,l]} \boldsymbol{n}\cdot\delta\boldsymbol{\epsilon} + \boldsymbol{m}\cdot\delta\boldsymbol{\kappa}\,\mathrm{dl} = -\int_{[0,l]} (\boldsymbol{A}\boldsymbol{\epsilon}(\boldsymbol{u}_0,\boldsymbol{0}))\cdot\delta\boldsymbol{\epsilon}\,\mathrm{dl} \; + \text{PBCs}, \quad \text{(beam formulation)} \tag{3.6b}$$

where, in Equation (3.6a), $\boldsymbol{\sigma}$ denotes the linearized Cauchy stress tensor, $\boldsymbol{\varepsilon}$ the linearized Green-Lagrange strain tensor, and $\boldsymbol{C}$ the Hooke material tensor. The problems defined in Equation (3.6) are solved under Periodic Boundary Conditions (PBCs) which is known, in the context of homogenization, to make the micro-macro-transition fulfill the Hill-Mandel condition. These PBCs can be enforced via multiple point constraints in the case of

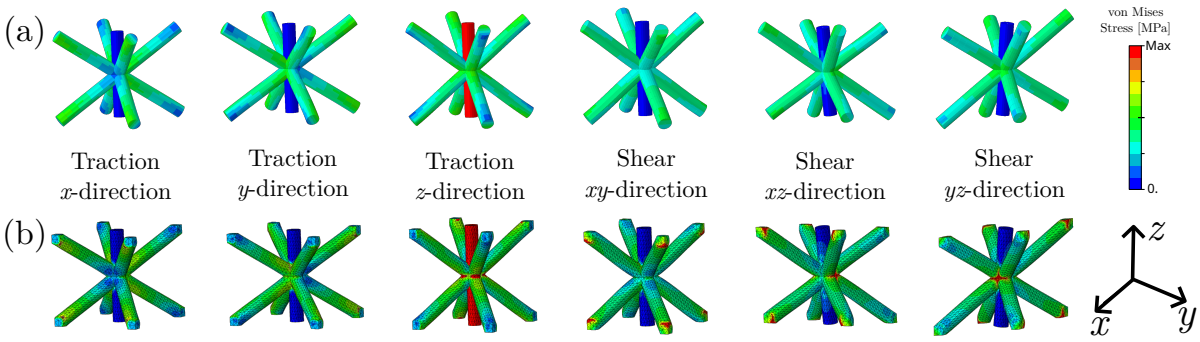

Fig. 3.7 von Mises stress related to the six loading cases for the BCCZ cell: (a) beam element model, and (b) 3D solid element model.

matching meshes at the periodic interfaces (Omairey et al. 2019). Other more advanced approaches, as for instance the Nitche method, might be used to enforce those PBCs for general non-conforming meshes (Henyš et al. 2019). Once the solution of Equation (3.6) in hand, one can compute effective (or average) stresses for the solid model $\hat{\boldsymbol{\sigma}}^{\text{solid}}$ or for the beam model $\hat{\boldsymbol{\sigma}}^{\text{beam}}$ as:

$$\hat{\boldsymbol{\sigma}}^{\text{solid}} = \frac{1}{|\Omega_0|}\int_{\Omega} \boldsymbol{\sigma}\, \mathrm{dv}, \qquad \hat{\boldsymbol{\sigma}}^{\text{beam}} = \frac{1}{|\Omega_0|}\sum_{i\in\mathcal{B}} \boldsymbol{n}^i \otimes \boldsymbol{x}_{\text{c}}^i, \tag{3.7}$$

where $\mathcal{B}$ denotes the set of boundary nodes.

Finally, the comparison between the beam models and the 3D solid model involves comparing the effective stresses. Here, this is done over six loading cases to account for the unit cell's behavior under normal and shear strains, as depicted in Figure 3.7. In other words, we choose in Equation (3.5), the macro-displacement gradient, and thus, the input displacement fields as:

$$\nabla\hat{\boldsymbol{u}}_0 = \frac{1}{2}\left(\boldsymbol{e}_k \otimes \boldsymbol{e}_l + \boldsymbol{e}_l \otimes \boldsymbol{e}_k\right) \quad \Rightarrow \quad \boldsymbol{u}_0^{kl}(\boldsymbol{x}) = \frac{1}{2}\left(x_l\boldsymbol{e}_k + x_k\boldsymbol{e}_l\right), \text{ for } \boldsymbol{x} \in \Omega_0, \tag{3.8}$$

where (k, l) take successively the following pairs of values: $\{11, 22, 33, 12, 13, 23\}$. By collecting the effective stresses for all six load cases, the accuracy of a beam model relative to a reference solid model is evaluated as:

$$\varepsilon_{\text{static}} = \frac{\|\mathbf{S}^{\text{solid}} - \mathbf{S}^{\text{beam}}\|_F}{\|\mathbf{S}^{\text{solid}}\|_F}, \tag{3.9}$$

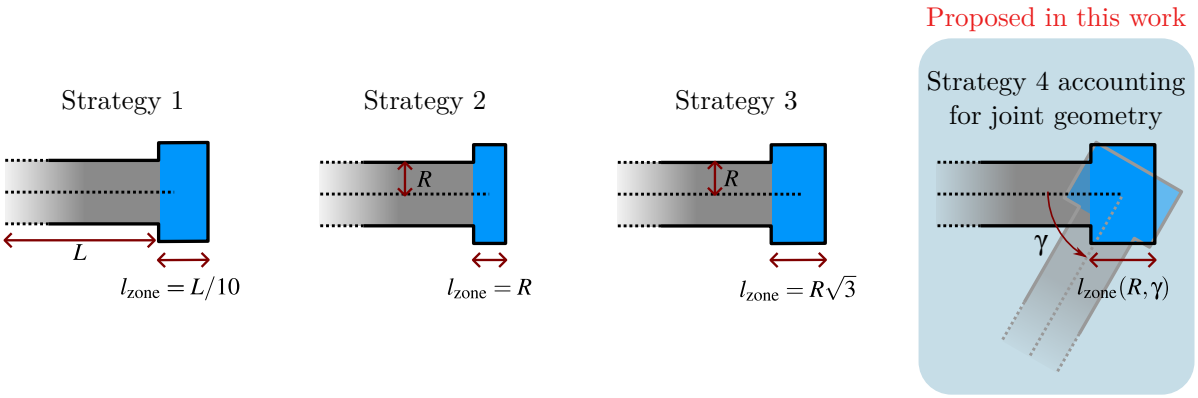

Fig. 3.8 Comparative schematic of three strategies found in literature: Strategy 1 (Labeas and Sunaric 2010), Strategy 2 (Gümrük and Mines 2013), Strategy 3 (H. Guo et al. 2020), and our proposed approach for l_{zone} .

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathbf{S} \in \mathbb{R}^{6\times 6}$ stacks the vectorized effective stresses (using Voigt notation for instance) for the six load cases:

$$\mathbf{S} = \begin{bmatrix} \hat{\sigma}_{11}^{11} & \hat{\sigma}_{11}^{22} & \dots & \hat{\sigma}_{11}^{23} \\ \hat{\sigma}_{22}^{11} & \hat{\sigma}_{22}^{22} & \dots & \hat{\sigma}_{22}^{23} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{23}^{11} & \hat{\sigma}_{23}^{22} & \dots & \hat{\sigma}_{23}^{23} \end{bmatrix}, \tag{3.10}$$

with $\hat{\sigma}_{ij}^{kl}$ being the ij-component of the effective stresses for the kl-load case. Operator $\mathbf{S}$ is the effective elasticity tensor in the context of first-order homogenization. In this study, the computation of the error $\varepsilon_{\text{static}}$ involves evaluating the differences between elasticity tensors obtained from a 3D solid model and a beam model. However, the focus is not directly on effective elasticity tensors. Instead, the error $\varepsilon_{\text{static}}$ is calculated to assess the accuracy of a beam model for a lattice unit cell, offering a relatively quick and general context for evaluation. The Frobenius norm was chosen as the accuracy evaluator due to its simplicity and effectiveness in providing a single, coherent indicator for comparing different models.

3.2.2 Defining the Length of the Stiffening Region

The parameter denoted l_{zone} (the length of the junctions) plays an important role in the numerical simulation, representing the size of the node, and consequently, its rigidity. Four strategies with different definitions of l_{zone} , drawn from the literature and investigated in this work, are detailed below and in Figure 3.8:

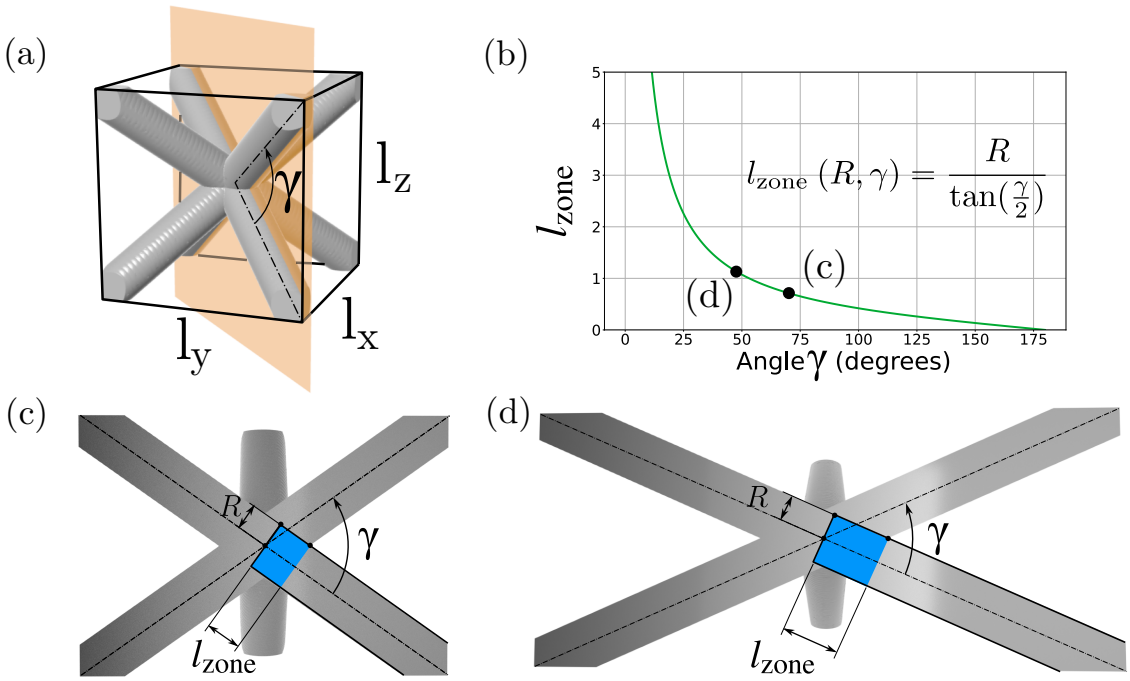

Fig. 3.9 (a) BCC unit cell half-cut by a center plane. (b) Graph illustrating the relationship between l_{zone} and the angle γ with $R = 0.5$ mm (Strategy 4). (c) Cross-sectional view of BCC unit cell ($l_x = l_y = l_z = 5$ mm, $R = 0.5$ mm which implies $\gamma = 70.5°$). (d) Cross-sectional view of BCC ($l_x = l_y = 8$ mm $l_z = 5$ mm, $R = 0.5$ mm which implies $\gamma = 47.7°$).

- Strategy 1: $l_{\text{zone}} = L/10$, with L being the length of the beam as done in Labeas and Sunaric (2010).
- Strategy 2: $l_{\text{zone}} = R$, with R being the radius of the beam according to Gümrük and Mines (2013).
- Strategy 3: $l_{\text{zone}} = R\sqrt{3}$, according to H. Guo et al. (2020).
- Strategy 4: $l_{\text{zone}}(R, \gamma)$ accounting for joint geometry hereafter.

Among strategies found in the literature, none seems to take into account the arrangement of the beams in the 3D space. However, the angle between the beams influences the overlap at the node, as highlighted in Figure 3.9. Indeed, a small angle implies a high nodal overlap. The length l_{zone} can be expressed regarding the radius R of the beams

and the angle γ through the following trigonometric relation:

$$l_{\text{zone}}\,(R,\gamma) = \frac{R}{\tan(\frac{\gamma}{2})}. \tag{3.11}$$

The angle γ is formed by the directional vectors of the neutral fibers of two adjacent beams (see Figure 3.9 again). In scenarios where more than two beams join at the same node, the smallest angle formed with other beams is considered for each beam. This relationship allows for defining potentially different l_{zone} for each beam along the structure. To simplify the description of the results, each strategy will be referred to by its previously associated number (see Figure 3.8).

3.2.3 Defining the Penalty Coefficient

Unlike the length of the junctions l_{zone}, it is less straightforward to determine an appropriate value for the parameter that penalizes the structural behavior in these regions. The penalty strategy favored in this work relies on increasing the radius instead of increasing the Young's modulus. This choice was motivated by its widespread use in recent literature (Gümrük and Mines 2013; H. Guo et al. 2020; Labeas and Sunaric 2010; Smith et al. 2013), but the same methodology can be applied if one chooses to penalize the elastic modulus.

The idea is to compute the error $\varepsilon_{\text{static}}$ on various lattice unit cells to identify an appropriate value. More specifically, we consider solving the following optimization problem:

$$\alpha_{\text{r}}^{*} = \underset{\alpha_{\text{r}} \in [1,5]}{\operatorname{argmin}}\ \varepsilon_{\text{static}}\,(\alpha_{\text{r}}), \tag{3.12}$$

for different unit cells. This optimization problem can be solved with the standard Brent optimization algorithm (Atkinson 1989; Brent and Watson 1973). This algorithm was employed in this work due to its straightforward usability and its gradient-free nature.

3.2.4 Evaluating the Effects of l_{zone} the Stiffening Region on Beam Model Accuracy

To effectively address the stiffness issue in the beam models, various strategies are evaluated in Figure 3.10, focusing on three unique geometric configurations. For reference, the different strategies considered to define the length of the stiffening region l_{zone} are summarized in Table 3.2.

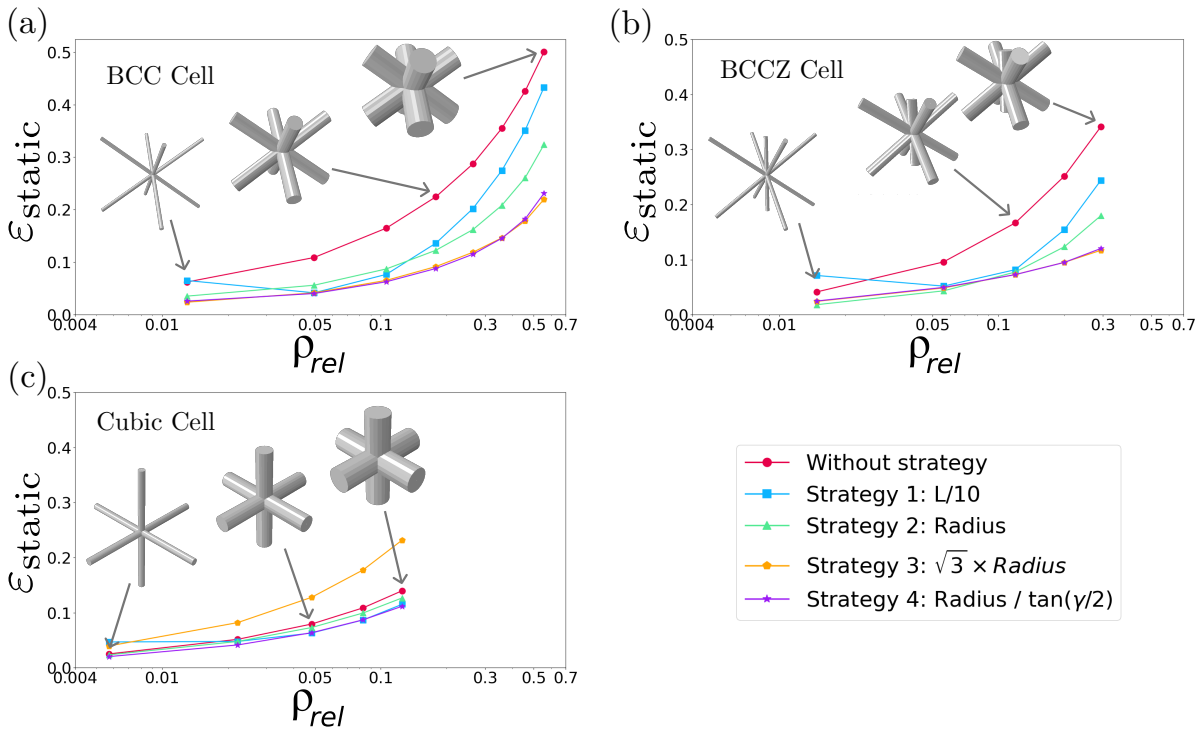

Fig. 3.10 $\varepsilon_{\text{static}}$ (relative error) as a function of ρ_{rel} (relative density) for multiple modelization strategies for (a) BCC, (b) BCCZ, and (c) Cubic unit cell.

For each strategy, the relative errors are computed within the range of relative density. In this section, the penalty coefficient is set arbitrarily at $\alpha_{\text{r}} = 1.5$. Figure 3.10(a) presents the results for the BCC unit cell. For the beam strategy without modification, the relative error rapidly grows as the relative density increases. This highlights the necessity for an approach that considers the concentration of material at the nodes when investigating lattice structures with less slender beams. The results of Strategy 1 are noteworthy, as the curve contains a minimum point situated at $\rho_{\text{rel}} = 0.05$. Moreover, at this density, the outcomes are equivalent to both Strategy 3 and Strategy 4. Thus, it becomes evident that Strategy 1 is only efficient with relative densities close to 0.05. While Strategy 2 offers fewer errors than the unmodified approach, Strategies 3 and 4 are the most optimal for this particular cell, yielding comparable results. This can be attributed to the fact that the angle is consistent for all beams in the BCC cell. In this case, Equation 3.11 delivers a result close to $R\sqrt{3}$, which corresponds to Strategy 3.

As shown in Figure 3.10(b), the various strategies lead to similar results for the BCCZ unit cell than for the BCC cell. It is noteworthy from the analysis that, within the context of low-density ranges, Strategy 2 exhibits enhanced efficacy for the BCCZ cell in comparison to the BCC cell. Strategies 3 and 4 prove once again to be equivalent and

Table 3.2 Summary of the different strategies for defining the stiffening region length l_{zone}

Strategy	**Definition of l_{zone}**	**Reference / Specificity**
Strategy 1	$l_{\text{zone}} = L/10$	Labeas and Sunaric (2010)
Strategy 2	$l_{\text{zone}} = R$	Gümrük and Mines (2013)
Strategy 3	$l_{\text{zone}} = R\sqrt{3}$	Based on geometric overlap, from H. Guo et al. (2020)
Strategy 4	$l_{\text{zone}} = \dfrac{R}{\tan\left(\dfrac{\gamma}{2}\right)}$	Takes into account the angle γ between connected beams

highly effective, for the same reasons as mentioned earlier. In contrast to the previously discussed cells, the Cubic cell is characterized by having fewer beams and exhibiting larger angles between these beams. The analysis of the Cubic cell depicted in Figure 3.10(c) shows that Strategy 4 performs the best, whereas Strategy 3 is not as effective compared to its performance in the previous cells. The errors generated by Strategy 3 are more significant than the model without any modification, as it overestimates the length to be modified. In this case, Equation 3.11 yields a modification length less than $R\sqrt{3}$. The strategy that consistently delivers stable results across different cell types and relative densities is Strategy 4, as it accounts for both the beam's radius and the angles within the structure. Therefore, our optimal methodology adopts Strategy 4.

3.2.5 Optimizing the Penalty Coefficient

Subsequently, the radius increase coefficient α_{r} was investigated for several cells. Strategy 4, which exhibited the highest performance previously, was selected for the length parameter modification l_{zone}. In Figure 3.11, the results indicate that optimizing the penalty parameter (*i.e.*, solving the optimization problem defined in Equation (3.12)) diminishes the errors between the solid and beam models throughout the full range of relative densities. This trend is consistently observed across all structural configurations studied.

The optimized coefficients α_{r}^* of all unit cells and spanning the entire range of studied relative densities, are presented in Figure 3.11(a). The optimal penalty coefficients α_{r}^* are grouped between 1.3 and 1.63 for all cells. The variations of the parameter across the range of relative density are different depending on the geometry. For instance, α_{r}^* changes little for the Octahedron cell, while it changes quite significantly for the BCCZ cell. Therefore, it seems not possible to determine a generalized relationship between the stiffening coefficient α_{r} and relative density ρ_{rel} for all cells. However, as the important

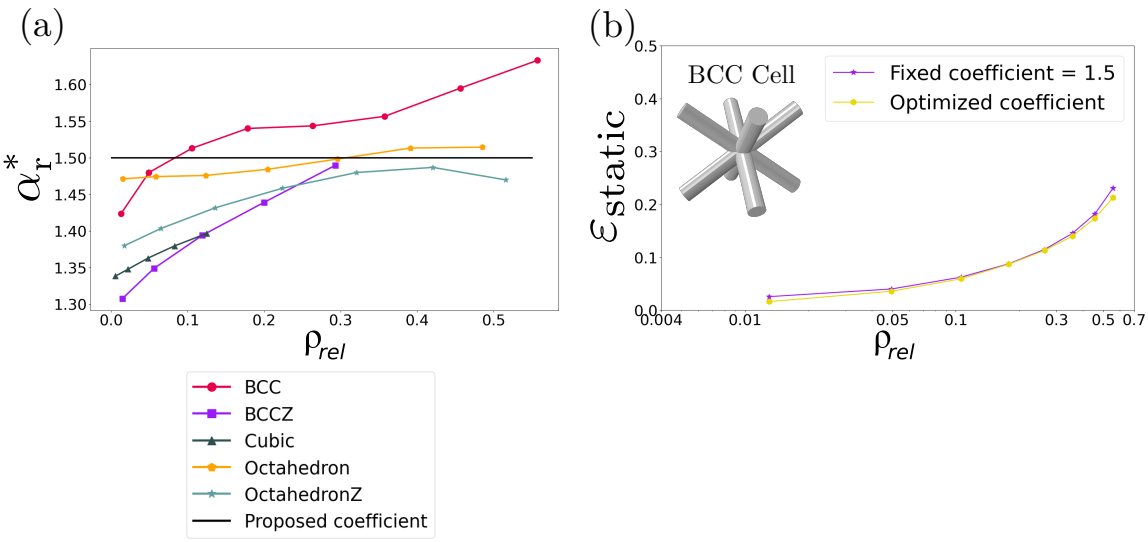

Fig. 3.11 Optimization result of α_r. (a) Aggregated results showing the optimized penalty coefficient α_r^* across various studied lattice structures. (b) Graph of relative error $\varepsilon_{\text{static}}$ versus relative density ρ_{rel} for a BCC unit cell employing both a fixed coefficient of 1.5 and an optimized coefficient.

thing is to reduce the highest relative errors $\varepsilon_{\text{static}}$, which correspond to cases of high relative density, it might be relevant to focus on these highest relative densities. The computed average of the optimal coefficients at maximum densities for each cell type is $\alpha_r = 1.496$ not far from $\alpha_r = 1.4$ as reported in existing literature (H. Guo et al. 2020; Labeas and Sunaric 2010; Smith et al. 2013). Thus, based on our results, we suggest setting the penalty parameter as $\alpha_r = 1.5$. Interestingly, as shown in Figure 3.11(b), such a fixed value leads to very similar results to the ones obtained with the optimal penalty parameters for the BCC cell. We also observed the same behavior for the other unit cells. Selecting l_{zone} as defined in Equation (3.11) and $\alpha_r = 1.5$ leads to satisfactory results with significant improvements to the original non-penalized beam models, as already highlighted in Figure 3.11(b). The deviations between beam models with the identified penalty method and 3D solid models is about 10% at different relative densities for each unit cell: 0.3 for BCC, 0.25 for BCCZ and 0.1 for Cubic cell.

3.3 Application of the Joint Stiffening Method to Dynamic Simulations

After establishing an optimal penalization strategy for static simulations, it is now essential to investigate whether the proposed beam model adjustments remain valid when

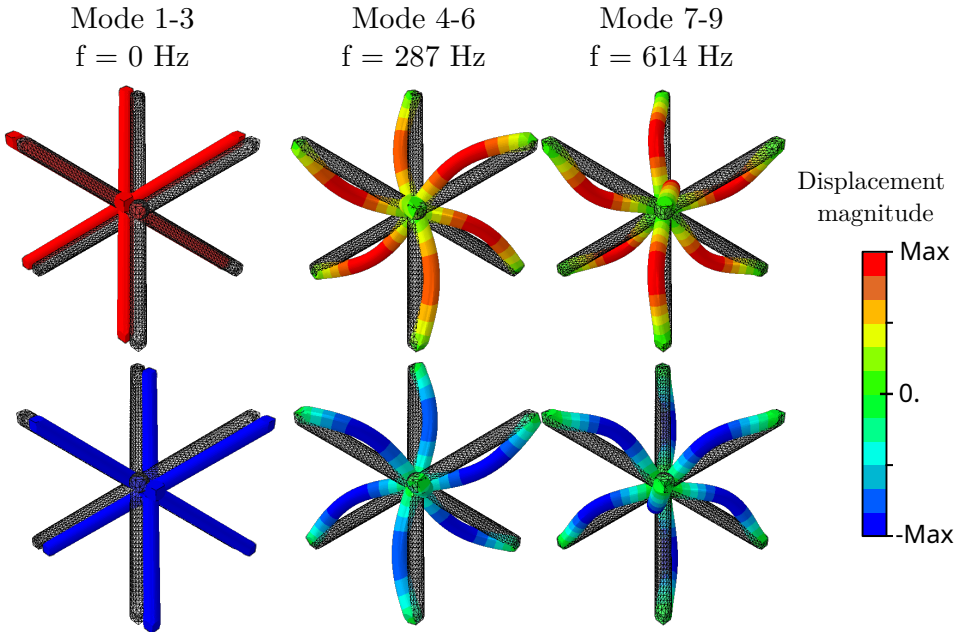

Fig. 3.12 The first nine vibrational modes of a modal analysis of BCC unit cell, each mode is tripled due to the geometry symmetry.

applied to dynamic analyses, where mass distribution and inertia effects play a critical role. The dynamic behavior of lattice structures is of particular interest for various applications, including vibration damping (J. Liu, K. Guo, et al. 2021; Scalzo et al. 2021) and high-speed impact response (Dwyer et al. 2023; Ling et al. 2019; Xiao et al. 2020; K. Yang et al. 2023). Despite their potential, only a limited number of studies have employed beam modeling approaches in such dynamic simulations. Scalzo et al. (2021) implemented a beam model with $l_{\text{zone}} = R$ and $\alpha_{\text{r}} = 1.4$ to assess the damping properties of BCCZ structures across various relative densities. F. Liu, L. Wang, et al. (2021) introduced a method to analyze the nonlinear frequency response by representing the joint as a spatial nonlinear spring-damper system with bilinear hysteresis. These examples highlight that beam models can provide a computationally efficient alternative to solid models for dynamic simulations, while still requiring careful treatment of nodal mass and stiffness adjustments. However, no studies, to our knowledge, have investigated the impact of the joint penalty method on the dynamic response of lattice structures.

3.3.1 Assessment of the Beam Models Accuracy via Modal Analyses over Unit Cells

In addition to the static analyses described in Section 3.2, it is valuable to compare the dynamic response of lattice structures obtained with beam models to the one obtained

with a 3D solid model. To achieve this, natural vibration analyses are performed over representative unit cells, *i.e.*, the solution involves:

$$-\omega^2 \int_\Omega \rho \boldsymbol{u} \cdot \delta \boldsymbol{u} \, \mathrm{dv} + \int_\Omega \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} \, \mathrm{dv} = 0 \; + \; \text{PBCs}, \qquad \text{(solid formulation)} \tag{3.13a}$$

$$-\omega^2 \int_{[0,l]} \mu \boldsymbol{u}_\mathrm{c} \cdot \delta \boldsymbol{u}_\mathrm{c} \, \mathrm{dl} + \int_{[0,l]} \boldsymbol{n} \cdot \delta \boldsymbol{\epsilon} + \boldsymbol{m} \cdot \delta \boldsymbol{\kappa} \, \mathrm{dl} = 0 \; + \; \text{PBCs}, \quad \text{(beam formulation)} \tag{3.13b}$$

where ρ and $\mu = \rho A$ denote the density and the linear density of the material, respectively. In this work, these natural vibration problems are performed under PBCs (as for the static analyses, see Equation (3.6)). This choice is motivated by the fact that these boundary conditions are often set in the context of (quasi)-periodic media, as is the case for lattice structures. Those natural modes could be observed within lattice structures. An example of results is given in Figure 3.12.[2]

When performing natural vibration analysis, a fixed number of modes n_modes is computed, specifically those associated with the smallest eigenvalues. This is done in this work, and the solid and beam models are compared by computing the following quantity:

$$\varepsilon_\mathrm{dyna} = \frac{\| \boldsymbol{t}^\mathrm{solid} - \boldsymbol{t}^\mathrm{beam} \|_2}{\| \boldsymbol{t}^\mathrm{solid} \|_2}, \tag{3.14}$$

where vector $\boldsymbol{t}$ collects the time periods:

$$\boldsymbol{t} = \begin{pmatrix} T_4 & T_5 & \dots & T_{n_\mathrm{modes}} \end{pmatrix}, \quad T_i = \frac{2\pi}{\omega_i}. \tag{3.15}$$

The first three modes are associated with translational rigid body motions (rotational rigid body are removed due to the PBCs), and thus $\omega_1 = \omega_2 = \omega_3 = 0$. We omit those modes that have infinity time periods. We also do not compare the frequencies (or pulsations) directly, as it leads to give more importance to the largest computed frequencies. On the contrary, employing the time periods in the expression of $\varepsilon_\mathrm{dyna}$ leads to prioritize the smallest computed frequencies.

[2] A supplementary video illustrating the dynamic response for this case is available at https://youtu.be/fybes6U17NQ.

3.3.2 Balance the Mass Distribution

The increase of the cross-sections near the joints compensates for the lack of rigidity at nodes, but it alters the mass distribution within the structure. At every node, the mass is increased due to the penalty coefficient α_{r} that augments the radius of the beam. Even if one chooses to penalize the Young's modulus, the density ρ_{zone} in the nodal zones can also be adapted to compensate for the overlaps near the junctions: beam models usually overestimate the mass of the structures. This structural modification needs to be considered for dynamic problems. More precisely, the objective is to define ρ_{zone} so that the mass of the beam model equals an objective mass m_{obj}, which is either the exact mass of the structure obtained with a computer-aided design (CAD) model (Terriault and Brailovski 2018), or an estimated mass, *i.e.*, we are looking for ρ_{zone} solution of:

$$\int_{\gamma_{int}} \rho A \,\mathrm{dl} + \int_{\gamma_{zone}} \rho_{\mathrm{zone}} A_{zone} \,\mathrm{dl} = m_{obj}, \tag{3.16}$$

where γ_{zone} is the penalized domain and γ_{int} is the internal domain (or the remaining domain), ρ is the material density, and A_{zone} is the area of the cross-sections in the penalized zones. In case of homogeneous material (and thus constant densities), the density ρ_{zone} can be defined as:

$$\rho_{\mathrm{zone}} = \frac{1}{V_{zone}}(m_{obj} - \rho V_{int}), \tag{3.17}$$

where V_{zone} and V_{int} are the volumes of the penalized and internal regions of the beam model, respectively.

3.3.3 Evaluating Added Mass Effects on the Structural Dynamics

In order to investigate the impact of the mass distribution within the structure, we perform modal analyses over unit cells and compute the dynamic error $\varepsilon_{\mathrm{dyna}}$ as defined in Equation (3.14). The number of computed modes is $N_{\mathrm{modes}} = 20$. Those analyzes are done with the following beam models:

- Beam model without any modifications (referred to as Model A).
- Beam model employing Strategy 4 with a radius increase coefficient $\alpha_{\mathrm{r}} = 1.5$ but no mass balance (referred to as Model B).

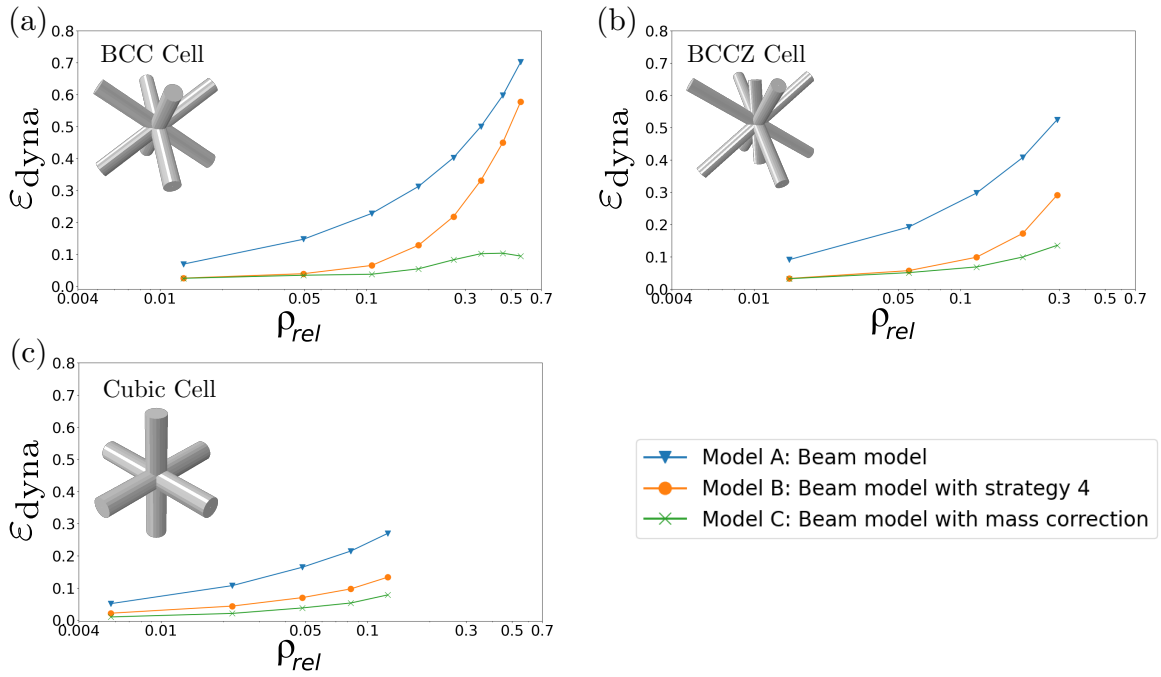

Fig. 3.13 Relative errors in frequency mode for (a) BCC, (b) BCCZ, and (c) Cubic cells across three modeling approaches showing the impact of joint modifications and mass distribution on accuracy.

- Beam model employing Strategy 4, $\alpha_r = 1.5$, and mass correction, detailed below (referred to as Model C).

In Figure 3.13 are compiled the results for BCC, BCCZ, and Cubic lattice geometries. Across all structures, a similar trend is observed for Model A: the relative error progressively increases with the relative density. This recurring trend highlights the essential need for modifications in beam joints to preserve accuracy. Model B reduces the natural frequency differences with the solid model at low relative density. However, this model exhibits a rapid escalation of the relative error as the relative density increases. This effect is attributable to, on one hand, the improvement of rigidity from the modifications at nodes, and on the other hand, the worsening of the mass repartition within the structure. Model C resolves this issue by re-establishing the mass distribution on the nodes. One can observe the large improvements obtained with this last model for all three lattice unit cells in Figure 3.13. In conclusion, conducting dynamic simulations of lattice structures using FE beam models requires careful consideration of joint behavior and mass dispersion that arises from the application of stiffening correction.

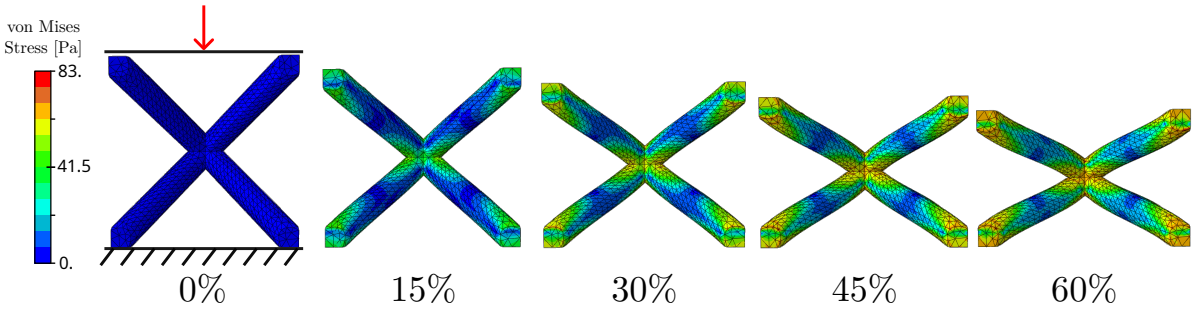

Fig. 3.14 Compression of a BCC unit cell with a 3D solid model.

3.4 Extension to Nonlinear Analyses and Experimental Comparison

In this section, we assess whether the improvements obtained with the selected penalty parameters in linear elastic simulations remain valid in more complex scenarios, such as geometrical and material nonlinearity, and further evaluate their relevance through comparison with experimental data.

3.4.1 Uniaxial Compression of a Unit-cell Made of Elasto-Plastic Material

To assess the viability of our joint stiffening methodology in more complex cases, we investigate uniaxial compression tests over unit cells using the elasto-plastic material model obtained in Section 3.1.3. Geometric nonlinearities are accounted for in the simulations, which allows for accurate modeling of large deformations and rotations (`NLGEOM` option in Abaqus). Several simulations are performed on the BCC unit cell presented in Figure 3.14. Indeed, three models are considered: a solid model considered as the reference model, a standard beam model without any joint stiffening methodology, and a beam model with our identified penalty method for the joint stiffening. BCC unit cell is studied with PBCs across a relative density range of 0.05 to 0.46.

Figure 3.15 illustrates the macroscopic stress-strain response from compression tests for each simulation approach. The standard beam model consistently underpredicts the structural response across all densities, confirming again the need to formulate a beam modeling approach suitable for FE simulations of lattice structures. The macro-stress is underestimated, which indicates that the model is not stiff enough (as expected). The modified beam model, however, aligns closely with solid FE results for geometries with a relative density below 0.3. Beyond this threshold, the modified beam model tends to

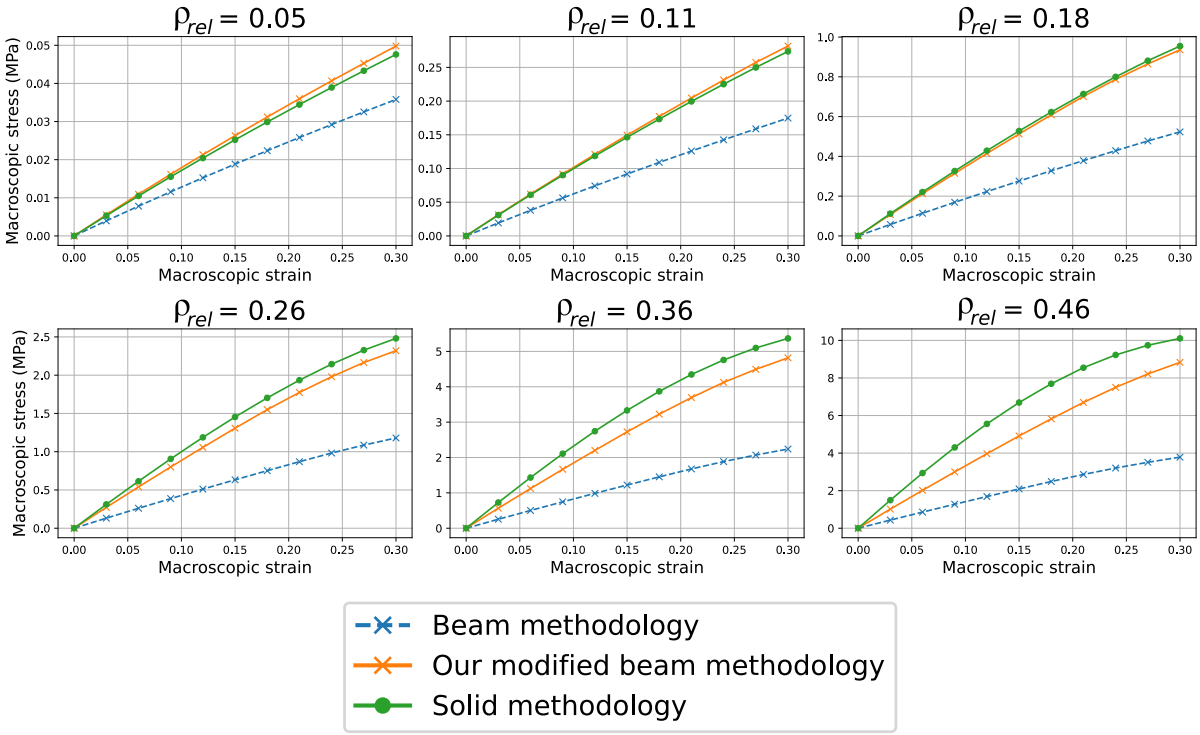

Fig. 3.15 Macroscopic stress-strain responses under uniaxial compression of the BCC unit cell, comparing the classical beam model, the proposed modified beam model, and the 3D solid reference model across different relative densities ρ_{rel} .

underestimate the macroscopic stress. These differences can likely be attributed to the mechanical behavior of the structures, which are not fully captured by the Timoshenko beam theory or the current treatment of plasticity within the beam formulation. It has been observed that plastic deformation initiates earlier in the solid simulation compared to the beam model. Moreover, the modified beam model shows a relocation of the maximum stress at the edge of the stiffened region, altering the beam's behavior. This effect is more pronounced in lattice structures with high relative density. In order to mitigate this type of stress relocalization, more advanced stiffness application strategies, such as progressive or smoothed transitions, could be envisaged. These aspects were not addressed in the present work and remain as perspectives for future investigations.

Thus, the beam FE simulation domain is enhanced thanks to our stiffening modification methodology for elasto-plastic cases too. However, limitations are observed in the model's capability to simulate with low error for high relative densities. The relative error varies depending on the geometry of the lattice structure (see Section 3.2.4). Therefore, the proposed strategy must be used cautiously to avoid reaching excessively high errors. The stiffening methodology demonstrates satisfactory accuracy for lattice structures

with a relative density below 0.3. Contrariwise, the methodology imposes no constraints on the geometries that can be treated, offering broad applicability in lattice structure simulations.

3.4.2 Experimental Validation of the Beam Penalization Methodology on 3D-printed BCC Structures

Finally, to extend beyond numerical model comparisons (beam versus solid), we conducted experiments on lattice structures. The lattices are composed of $6\times6\times6$ square cells with a side length of 10 mm (see Figure 3.16). Two relative densities were chosen: one in the zone where the beam simulation method performs correctly (beam radius $R = 1.0$ mm $\Rightarrow \rho_{\text{rel}} = 0.169$) and one with a relative density too high (beam radius $R = 1.5$ mm $\Rightarrow \rho_{\text{rel}} = 0.345$). Quasi-static compression tests were conducted experimentally on these lattice structures, and similar tests were simulated using beam models without and with our joint stiffening approach, as well as solid FEM simulations.

The lattice specimens used for the experimental validation were manufactured using the same material jetting process as the tensile samples described in Section 3.1.3. Quasi-static compression tests were performed on the printed specimens using the same Lloyd LR 50K universal testing machine as for the tensile tests. The experiments were conducted under displacement control at a constant crosshead speed of 1 mm/min, ensuring that the loading rate remained within the quasi-static regime. The boundary conditions were imposed through flat compression plates with polished surfaces to ensure homogeneous loading. Table 3.3 summarizes the main parameters of the manufacturing and testing protocols.

Figure 3.17 illustrates the global correlation between the 3D-printed specimens during testing and the simulations, including both our modified beam methodology and the solid FEM simulation, before major damage to the experimental structure. In the case of full-scale lattice structure simulations, the number of elements in the mesh for solid models increases significantly, leading to a substantial computational burden. To manage this, we applied symmetry in the X, Y and Z planes to reduce the number of mesh elements. This constraint was not necessary for the beam models. Additionally, the simulation time for solid models is considerably longer, requiring over 14 hours compared to just 2 minutes for the beam models on a standard laptop, excluding the time needed to generate the geometry and mesh. A supplementary video illustrating the compression

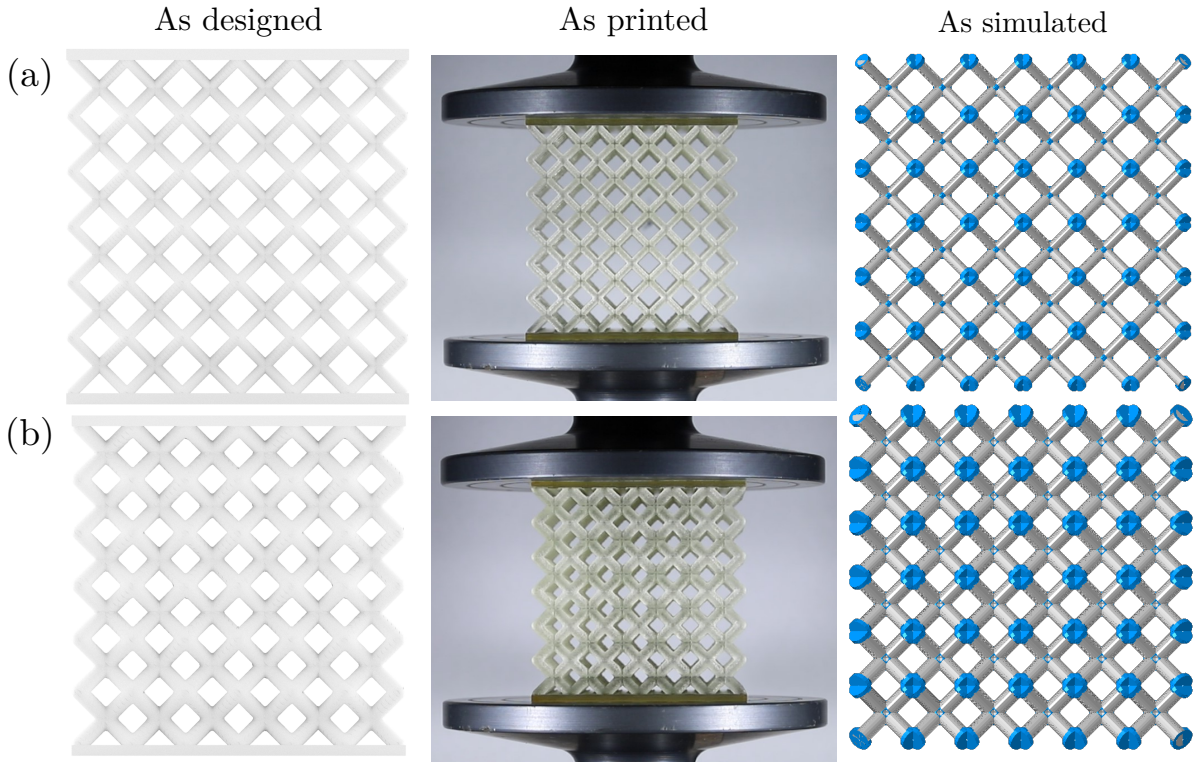

Fig. 3.16 As-designed, as-printed, and as-simulated BCC lattice specimens prepared for compression tests (Beam radius: (a) 1 mm and (b) 1.5 mm).

Table 3.3 Summary of experimental parameters for manufacturing and compression testing of lattice specimens

Parameter	**Description**
3D Printing	
Printer	Stratasys Objet260 Connex3
Material	VeroClear™ photopolymer
Process	PolyJet
Nozzle diameter	30 μm
Layer thickness	30 μm
Resolution	600 dpi
Compression Test	
Machine	Lloyd LR 50K
Load cell capacity	50 kN
Crosshead speed	1 mm/min
Plates	Flat and polished

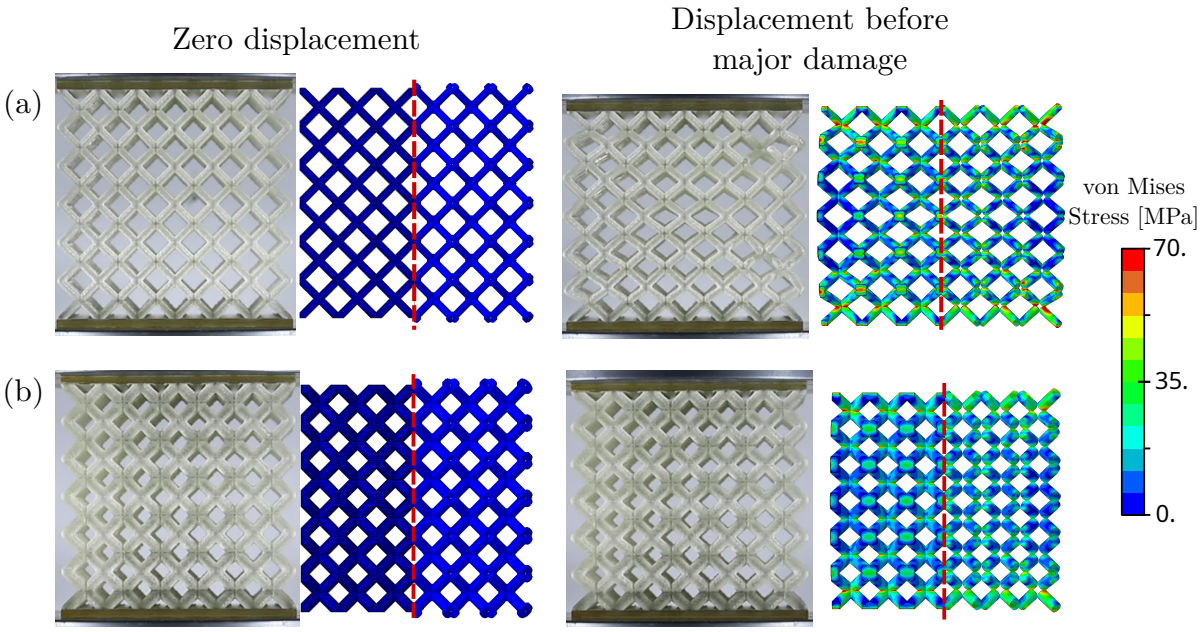

Fig. 3.17 Comparative visualization of 3D-printed specimens and simulation results (Solid FEM on the left of the red line and beam with optimal penalty parameters on the right) before major damage on experimental specimen (strain of (a) 0.07 and (b) 0.03) for BCC at relative density of (a) 0.17 and (b) 0.34.

test of the BCC lattice structures is provided[3]. The results exhibit a good concordance, underscoring the accuracy of the simulation in replicating experimental findings.

Figure 3.18 synthesizes the macroscopic stress-strain data for each structural characterization method. The first point to note is that the differences between solid and beam models are consistent with those observed in unit-cell simulations (Subsection 3.4.1). For the BCC lattice structure with a 1 mm strut radius (Figure 3.18(a)), our modified beam methodology accurately aligns with the experimental curve for small deformations, whereas the unmodified beam method underestimates the stiffness of the structure. At higher deformations, localized strut failures within the structure emerge, which are not accounted for in the material behavior simulations, leading to discrepancies. A more sophisticated material model would be required to accurately capture these phenomena; however, this is beyond the scope of this work. In the case of the BCC structure with a 1.5 mm strut radius (Figure 3.18(b)), the unmodified beam model performs very inadequately. The penalized beam model performs better, but the differences with the experimental results are not negligible. Interestingly, the solid model also fails to accurately predict the mechanical behavior of the structure, even if it performs a bit

[3]Available at https://youtu.be/zrHPoNpuriY.

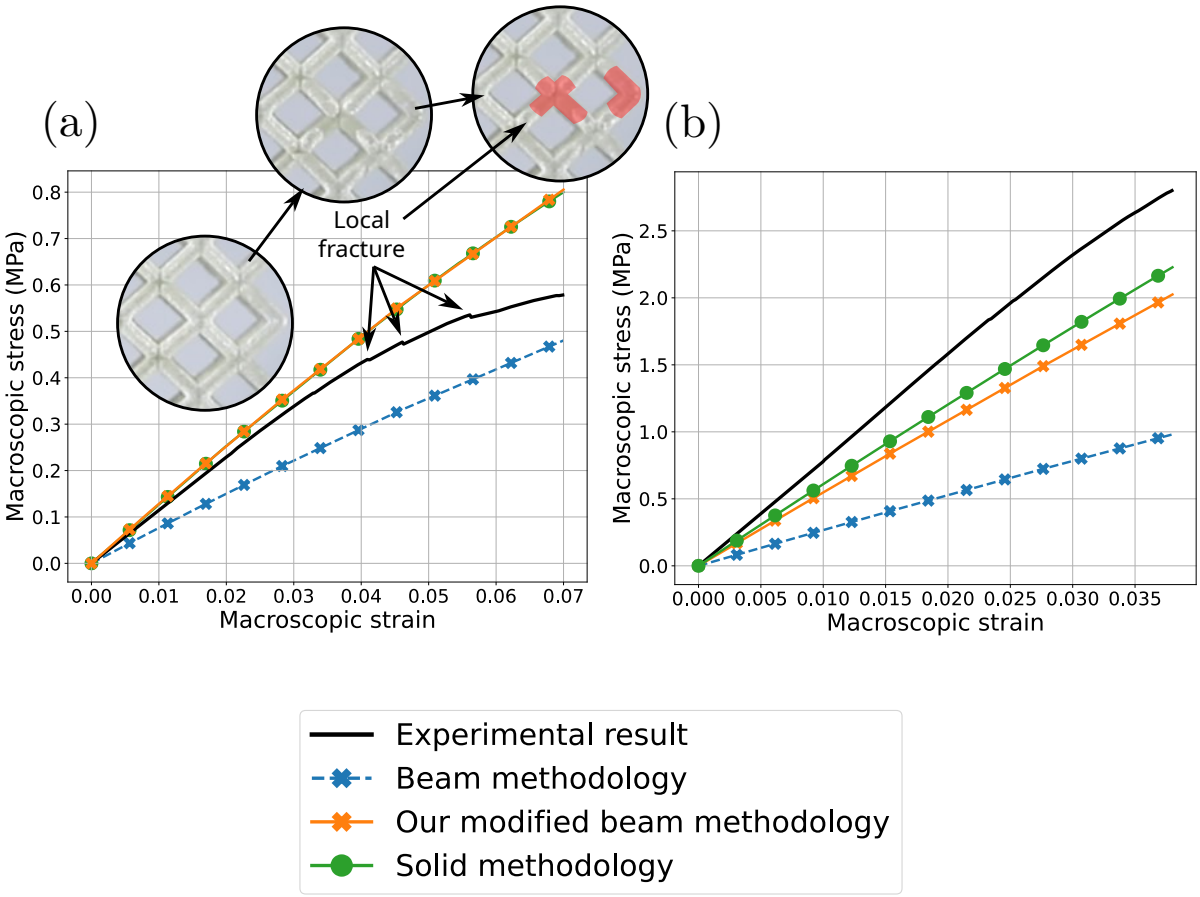

Fig. 3.18 Macroscopic stress-strain results: Experimental, Solid, Timoshenko, and our modified beam methodology for BCC at relative densities ρ_{rel} of (a) 0.17 and (b) 0.34.

better. This is likely due to an insufficiently accurate material model, resulting in these discrepancies.

3.5 Conclusion on the Beam Model Penalization Methodology

In the context of beam-based modeling of lattice structures, accurately capturing the joint stiffness remains a major challenge, especially for complex geometries and diverse relative densities. In this chapter, a novel methodology for simulating lattice structures using a FE beam model is proposed (See Figure 3.19). By employing Timoshenko beam elements, this methodology significantly reduces the computational expenses commonly associated with solid FE simulations. This strategy addresses the lake of rigidity at the

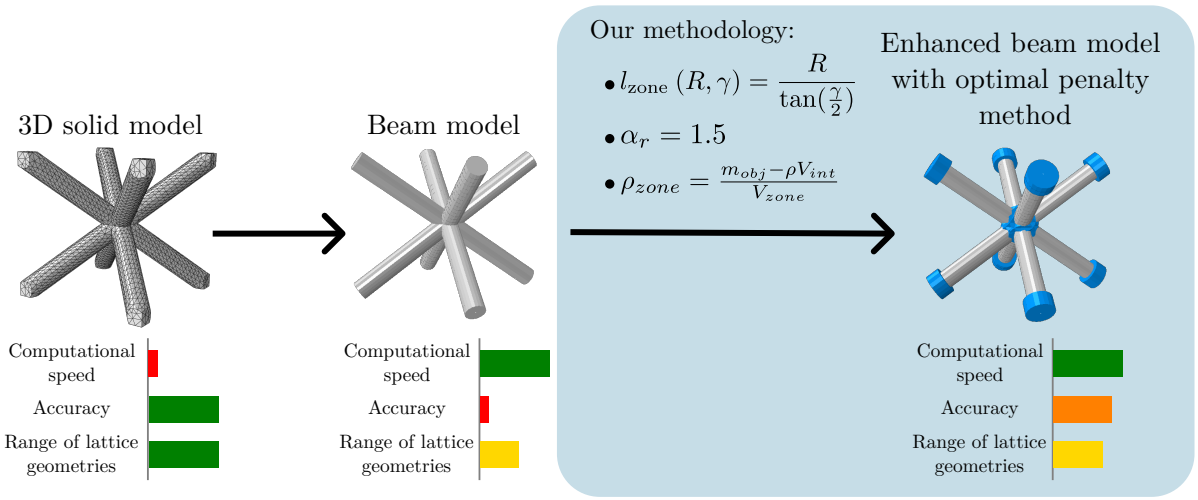

Fig. 3.19 Overview of the optimal penalty method for the joint stiffening in beam models.

nodes of the structure using three critical parameters:

$$l_{\text{zone}}(R,\gamma) = \frac{R}{\tan(\frac{\gamma}{2})}, \qquad \alpha_{\text{r}} = 1.5, \qquad \rho_{\text{zone}} = \frac{1}{V_{zone}}(m_{obj} - \rho V_{int}). \tag{3.18}$$

This approach is based on a geometrical definition of the nodal stiffening length l_{zone}, which is dependent on the radius R of the beam and the angle γ between the beams, allowing to account for the geometry of each junction in the structure. The penalization coefficient α_{r} is optimized to minimize the relative error between the beam and solid models for a range of relative densities, leading to a value of $\alpha_{\text{r}} = 1.5$. Finally, the mass distribution in the penalized regions is adjusted via ρ_{zone} to ensure that the beam model accurately represents the mass of the structure, which is crucial for dynamic simulations. However, limitations still exist on the domain of relative densities. The stiffening methodology demonstrates satisfactory accuracy for lattice structures with a relative density below 0.3. Experimental testing further validates the accuracy of the modified beam model in predicting the behavior of entire lattice structures.

The proposed simulation strategy is applicable not only to known lattice structures, but also to geometries with varying cellular dimensions or relative density gradients. The angles between the beams vary depending on the gradient function used, which is not considered in strategies using a constant l_{zone}. Functionally graded lattice structures are recognized for their superior performance across a broad spectrum of applications, especially in terms of energy absorption (See Section 2.1.1).

Another branch of research is focusing on the development of more advanced beam models that can capture geometric discrepancies from 3D models and account for more

complex deformation behaviors, such as swelling or inelastic deformations (Alzate Cobo et al. 2025; Schommartz et al. 2024; Weeger, Valizadeh, et al. 2023). While these approaches can potentially offer improved accuracy, they often require substantial modifications to the numerical formulations and solver architectures, making them intrusive and challenging to integrate into standard commercial FE software such as Abaqus. In contrast, the penalty-based method proposed in this work is fully non-intrusive and easily implementable. It operates solely through geometric parameter adjustments, without altering the underlying FE formulations, which makes it particularly suitable for practical applications.

This novel methodology enables the efficient determination of the mechanical properties of a lattice structure. It constitutes the foundational step toward developing a strategy for identifying the optimal lattice geometry for a specific application.

Chapter 4

Domain Decomposition Enhanced by Surrogate Model for the Efficient Simulation of Lattice Structures

"Dans la vie, rien n'est à craindre, tout est à comprendre."
— Marie Curie

Contents

4.1 Foundations for Exploring Locally Tailored Lattice Structures . 92
4.1.1 Hybrid Lattice Geometry Paving a Large Design Space 93
4.1.2 Benchmark Structural Configurations 95
4.2 Domain Decomposition Method Applied to Lattice Structures . 97
4.2.1 Global overview of DDM 98
4.2.2 Primal interface problem 102
4.2.3 Solving the Interface Problem 104
4.2.4 Application to Beam-like Lattice Structures 106
4.3 Surrogate Model for Fast Analysis 110
4.3.1 Reduced Basis via Greedy Algorithm 111
4.3.2 A variety of Surrogate Modeling Techniques 115
4.3.3 Model Behavior and Accuracy Assessment 119
4.4 Embedding Physical Knowledge into Neural Network 126
4.4.1 Embedding Physics for Efficient and Generalizable Surrogate Models 127
4.4.2 Training Strategy for the PANN 134
4.4.3 Efficient Dataset Generation 137
4.4.4 Training Procedure and Evaluation of the PANN 141
4.4.5 Nonlinear CG Solver for PANN-Based DDM 145
4.5 Discussion and Conclusion 148

4.1 Foundations for Exploring Locally Tailored Lattice Structures

The objective is to develop a versatile framework that can accommodate various types of unit cells, optimization parameters, and structural patterns within the design space, as illustrated in Figure 4.1. The framework must be applicable to any type of unit-cell modeling, whether based on 3D solid, shell, or beam representations, as illustrated in Figure 4.1(a). Moreover, the accessible design space should be as extensive as possible, not only at the cell scale, as shown in Figure 4.1(b), but also at the structural scale, by enabling graded configurations or sweep patterns, as depicted in Figure 4.1(c). The following section introduces the design space and benchmark configurations that will serve as references for the studies presented in this chapter.

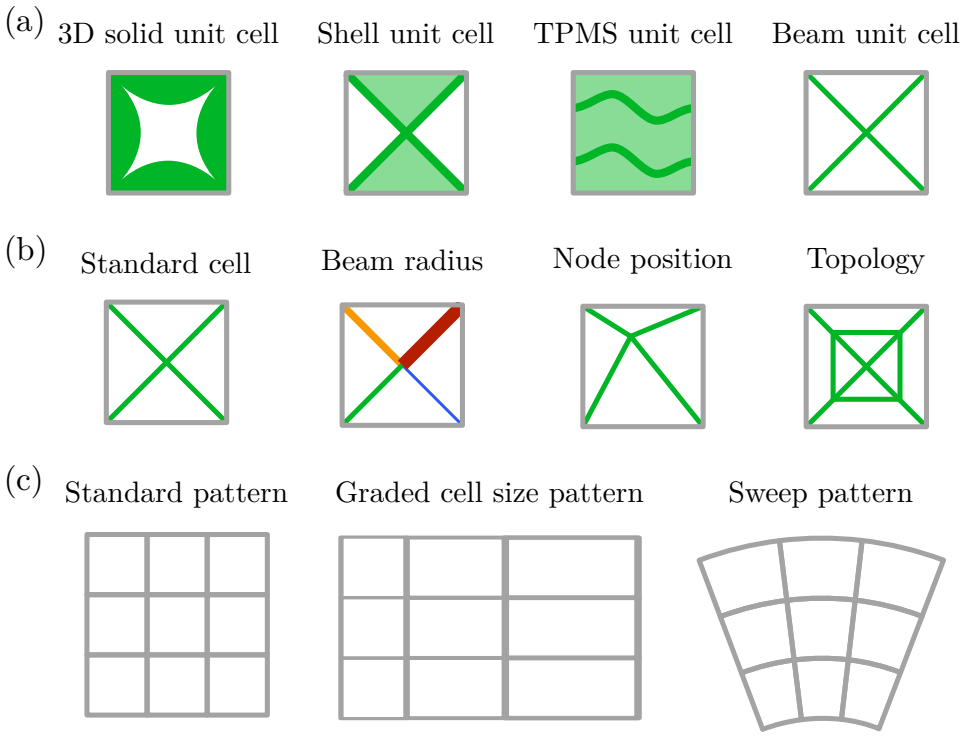

Fig. 4.1 Illustration of the versatility of the proposed method in terms of (a) unit cells, (b) optimization variables, and (c) structural patterns within the design space.

4.1.1 Hybrid Lattice Geometry Paving a Large Design Space

The simulation method presented in the following work is defined at the cell level; consequently, the parametrization procedure must also be established at this same scale. The parametrization method has a significant impact on the attainable diversity of mechanical behaviors. Increasing the number of parameters in the design space potentially enables more complex responses; however, it also increases the overall complexity of the problem, thereby making the optimization process more challenging. This section presents a method for selecting design parameters in a way that maximizes mechanical diversity within the design space while avoiding an excessive increase in complexity. Since an enhanced beam simulation method was developed in Chapter 3, truss lattice structures constitute a natural choice for testing the proposed methodology. However, as illustrated in Figure 4.1(a), the framework is not limited to this type of geometry.

The most commonly used parameter for truss lattice structures is the radius of the struts composing each unit cell, as it is straightforward to implement. However, the variety of mechanical behaviors that can be achieved with a single beam radius remains limited. To overcome this limitation, the concept of hybrid lattice structures has emerged as a promising solution. Hybrid lattice structures combine multiple base geometries, each with its own parameterization, thereby creating a more diverse and adaptable design space. This concept has been explored in various studies, encompassing both 2D and 3D configurations, as well as truss- and surface-based lattice structures (Costa et al. 2022; Tancogne-Dejean et al. 2018; J. Wang, J. Zhu, et al. 2023; M. Zhao, X. Li, D. Z. Zhang, et al. 2023). Although most of these works use two or three primitive unit cells, J. Liu, Zou, et al. (2025) extend this approach by employing eight different truss geometries to generate a wide range of mechanical behaviors, which are subsequently used in a TO context. Moreover, the concept of hybrid lattice structures is not limited to mechanical performance alone, as it can also be employed in the design of acoustic metamaterials (L. Li et al. 2025). The principal advantage of hybrid lattice structures lies in their ability to combine the strengths of distinct, primitive unit cells. For instance, Surjadi et al. (2025) demonstrated that integrating classical stiff truss geometries with softer woven geometries leads to enhanced energy dissipation properties. A step further to augment the design space is to relax the concept of primitive unit cells by developing a design methodology that independently controls both the topology and the radius of each beam (Zheng et al. 2023). However, this approach significantly increases the complexity of the design space, as the number of parameters to optimize grows rapidly, thereby making the exploration and optimization processes more challenging.

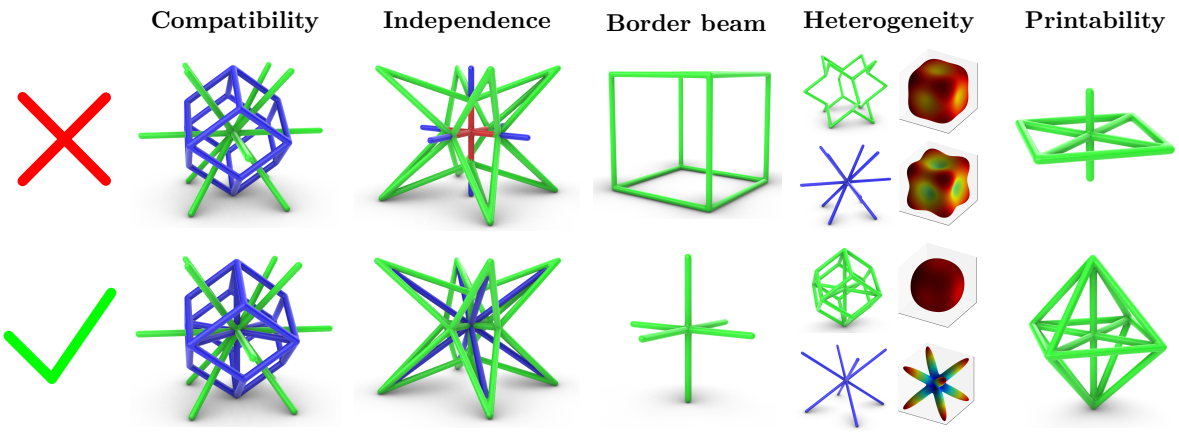

Fig. 4.2 Selection criteria for the base geometries used in the construction of hybrid lattice cells. Green beams represent geometry 1, and blue beams represent geometry 2. Red beams indicate regions where both geometries overlap.

The hybrid lattice structures approach aims to balance design flexibility and computational efficiency. In parametric optimization, reducing the number of parameters improves computational efficiency. However, achieving a wide diversity of mechanical behaviors is crucial to allow the optimization algorithm to effectively explore complex design spaces. In order to construct a basis of hybrid geometries that is as diverse as possible, a set of criteria is established for the selection of primitive unit cells, as summarized in Figure 4.2 and described hereafter:

- **Compatibility**: Each geometry must be connected to at least one common node to ensure structural integrity (this criterion could be relaxed to allow interpenetrating cells, but this would require a more advanced simulation method to account for contact interactions between beams).
- **Independence**: No beam should simultaneously be part of multiple geometries to maintain a distinct and manageable parameterization.
- **Absence of boundary beams**: No beam should coincide with the cell boundary. When a beam lies on the boundary of a cell, it introduces boundary conditions that are more complex to handle within our Domain Decomposition Method (DDM) framework and are therefore not considered in this study.
- **Heterogeneity**: The selected geometries must exhibit diverse mechanical behaviors to maximize design versatility.
- **Printability**: Each cell geometry must be individually manufacturable, ensuring that all unit cells can be printed independently using the selected AM process.

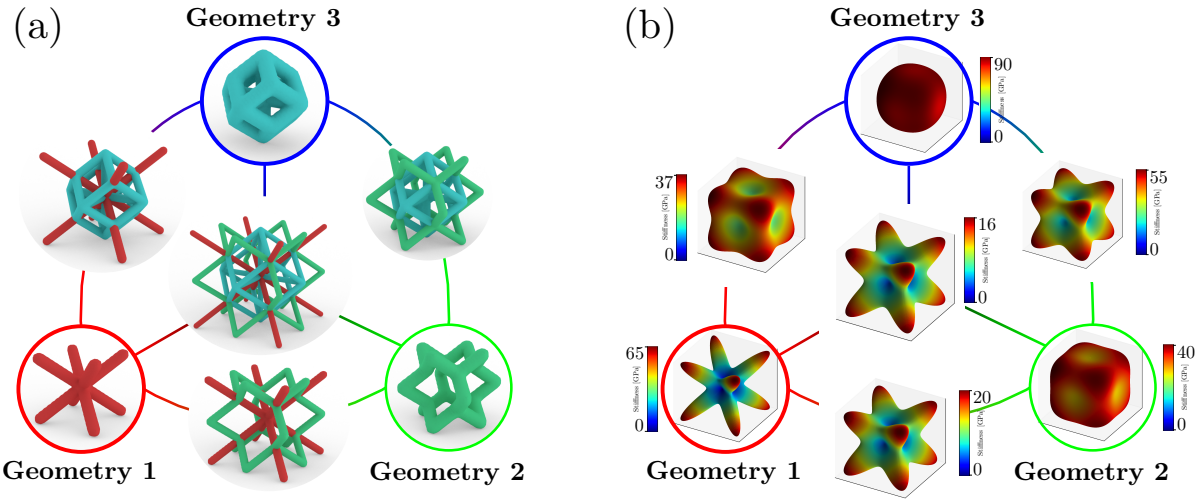

Fig. 4.3 Base geometries selected for constructing hybrid lattice structures. (a) Geometric representation of the three individual truss configurations, and (b) corresponding homogenized properties.

In our approach, we have selected three base geometries that satisfy the selection criteria, as illustrated in Figure 4.3. A key advantage of hybrid lattice structures is that combining multiple anisotropic base geometries can yield nearly isotropic mechanical responses (Tancogne-Dejean et al. 2018). However, a recurring issue when using multiple cell geometries within a single structure is ensuring geometric connectivity between adjacent cells (Ferro et al. 2022). In the present study, this difficulty is addressed by limiting the beam radius of each geometry to a small value. This choice offers a dual benefit: it ensures geometric compatibility between adjacent cells by facilitating their connection, and simultaneously prevents abrupt variations in stiffness that would otherwise hinder the convergence of gradient-based optimization algorithms, as further detailed in Chapter 5.

4.1.2 Benchmark Structural Configurations

To enable an efficient and meaningful comparison of the different types of simulation methods considered in this work, we define a set of case studies that vary in both structural configuration and loading conditions. These configurations are selected to span a diverse set of parameters, including the number of cells and the type of mechanical solicitation. The first study case corresponds to a cantilever L-beam structure composed of 27 cells, subjected to a vertical load applied at the lower tip of the beam. The second case is a three-point bending experiment, a standard procedure used to evaluate the stiffness and strength of materials and structural members. The three-point bending

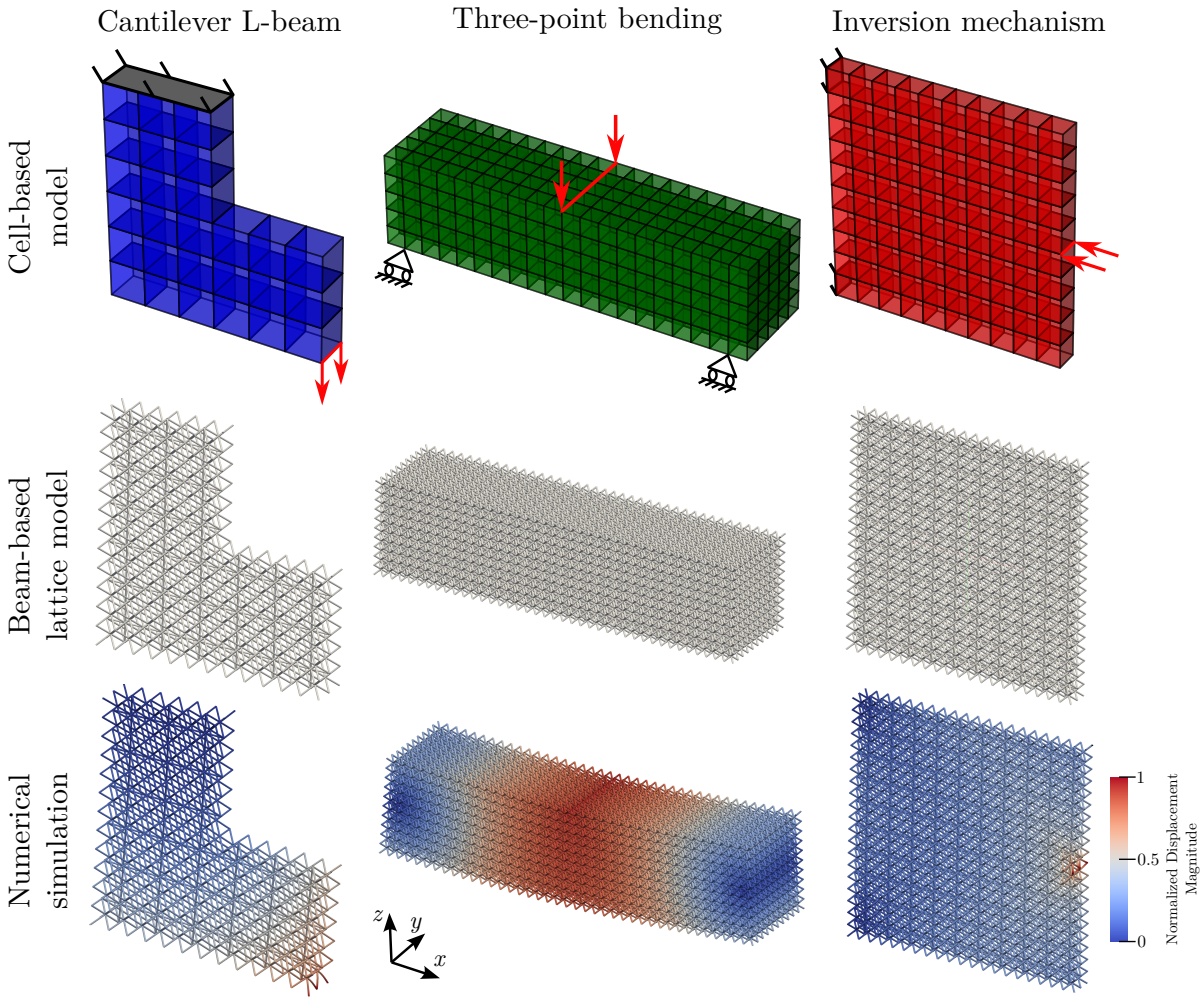

Fig. 4.4 Illustration of the three structural configurations used as benchmark problems. Each cell of $1 \times 1 \times 1$ mm is populated with a hybrid geometry, and the beam radius is set to 0.05 mm for the three base geometry. For each problem, results are shown using the cell-based model, the beam-based lattice model, and full numerical simulations.

configuration consists of a thick beam supported by two support lines located beneath the beam. Due to the presence of a symmetry plane at the mid-height of the beam, only half of the structure is simulated, which significantly reduces the computational cost. The third study case is formulated as an inversion mechanism, where the structure is subjected to a compressive force at the center of the right edge, while the opposite edge corner is constrained at the top and bottom. This configuration has been specifically selected to be compatible with optimization procedures targeting the design of an compliant mechanism. Each of these configurations, along with the corresponding FE simulation results, is illustrated in Figure 4.4. Their main characteristics, including geometric layout, and computational cost, are summarized in Table 4.1.

Table 4.1 Main characteristics and computation times of the three benchmark cases used for surrogate model evaluation. Computation time is split into structure generation and system resolution.

	Cantilever L-beam	**Three-point bending**	**Inversion mechanism**
No. of Cells	27	$20 \times 5 \times 5 = 500$	$10 \times 10 \times 1 = 100$
No. of DOFs	106,782	981,294	197,406
Symmetry	None	Mid-length plane (yz)	Mid-height plane (xy)
Computation time (s)	4.05 [0.25, 3.8]	506.2 [2.65, 503.55]	14.6 [0.52, 14.08]
Benchmark purpose	Simple domain	Large structure	Optimization framework

4.2 Domain Decomposition Method Applied to Lattice Structures

In a design space where each cell is treated independently, meaning that each cell is defined by its own set of geometric parameters, the dimensionality of the design space increases with the number of cells within the structure. This property is fundamentally different from traditional strategies such as global parametrization or functionally graded structures. Conventional FEM approaches are capable of accurately simulating the behavior of such lattice structures. However, within an optimization framework, evaluating the gradient of each parameter quickly becomes computationally demanding. It is therefore essential to develop a methodology that is both flexible and computationally efficient.

The intrinsic quasi-periodicity of lattice structures makes them well-suited for the application of DDM, especially non-overlapping approaches (Gosselet and Rey 2006). Instead of solving a single large system, the computational domain is partitioned into non-overlapping subdomains, where local problems are solved independently, as discussed in Section 2.2.2 This section introduces the theoretical background and the practical implementation of the DDM framework in the context of lattice structures.

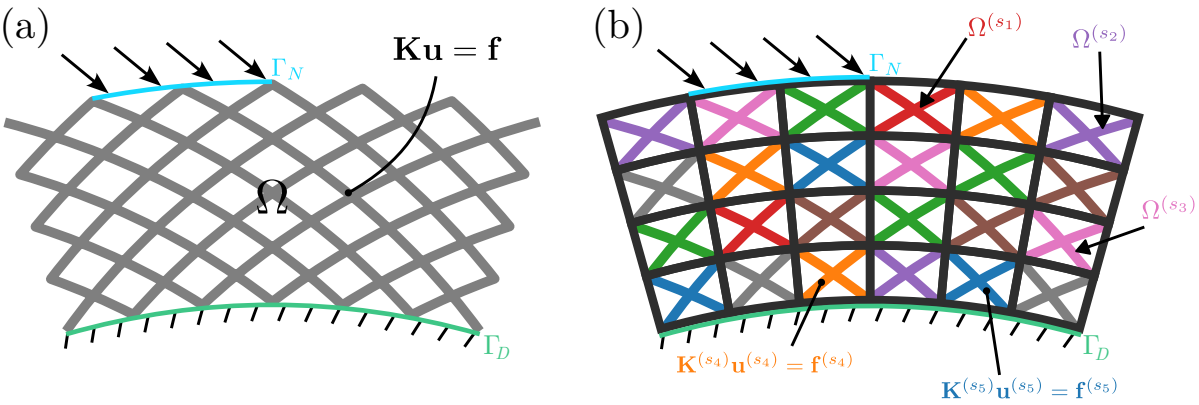

Fig. 4.5 Schematic representation of the DDM applied to a lattice structure. (a) Full-scale FEM formulation with the global equilibrium equation $\mathbf{Ku} = \mathbf{f}$, and (b) decomposition into cell subdomains $\Omega^{(s_i)}$ with corresponding local equilibrium equations $\mathbf{K}^{(s_i)}\mathbf{u}^{(s_i)} = \mathbf{f}^{(s_i)}$.

4.2.1 Global overview of DDM

Decomposing the computational domain

In the standard linear elasticity context, the problem is formulated as:

$$\begin{cases} \operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{b}_{ext} = 0 & \text{in } \Omega, \\ \boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{over } \Gamma_D, \\ \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{t}_{ext} & \text{over } \Gamma_N, \end{cases} \tag{4.1}$$

where Γ_D and Γ_N denote the boundary domains subject to prescribed displacements $\boldsymbol{u}_0$ (Dirichlet conditions) and body forces $\boldsymbol{b}_{ext}$ and tractions $\boldsymbol{t}_{ext}$ (Neumann conditions), respectively (see Figure 4.5(a)).

In the context of non-overlapping DDM, this problem is reformulated by decomposing the domain into non-overlapping subdomains $\Omega^{(s)}$, where local problems are solved independently and the continuity across subdomains is enforced through interface constraints, illustrated in Figure 4.5(b). For each subdomain (s), the local problem is as follows:

$$\begin{cases} \operatorname{div}(\boldsymbol{\sigma}^{(s)}) + \boldsymbol{b}_{ext} = 0 & \text{in } \Omega^{(s)}, \\ \boldsymbol{\sigma}^{(s)} = \boldsymbol{C}^{(s)} : \boldsymbol{\varepsilon}(\boldsymbol{u}^{(s)}) & \text{in } \Omega^{(s)}, \\ \boldsymbol{u}^{(s)} = \boldsymbol{u}_0 & \text{over } \Gamma_D \cap \partial\Omega^{(s)}, \\ \boldsymbol{\sigma}^{(s)}\boldsymbol{n}^{(s)} = \boldsymbol{t}_{ext} & \text{over } \Gamma_N \cap \partial\Omega^{(s)}. \end{cases} \tag{4.2}$$

In order to recover the linear elasticity problem (4.1) posed over the full domain Ω, kinematic compatibility and equilibrium of the tractions along the subdomain interfaces Γ should be added to the local problems (4.2). For every couple (s, s') of neighboring subdomains, these two coupling conditions can be defined as:

$$\boldsymbol{u}^{(s)} - \boldsymbol{u}^{(s')} = 0 \qquad \text{over} \quad \Gamma^{(s,s')}, \tag{4.3a}$$

$$\boldsymbol{\sigma}^{(s)}\boldsymbol{n}^{(s)} + \boldsymbol{\sigma}^{(s')}\boldsymbol{n}^{(s')} = 0 \quad \text{over} \quad \Gamma^{(s,s')}, \tag{4.3b}$$

with $\Gamma^{(s,s')} = \partial\Omega^{(s)} \cap \partial\Omega^{(s')} \subset \Gamma$.

Applying the FEM to the coupled system of equations (4.2) and (4.3) leads to the follwing linear systems to be solved:

$$\begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \begin{pmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_i^{(s)} \\ \mathbf{f}_b^{(s)} \end{pmatrix}, \tag{4.4a}$$

$$\sum_s \underline{\mathbf{B}}^{(s)} \left(\mathbf{u}_b^{(s)} - \mathbf{u}_{0,b}^{(s)} \right) = \mathbf{0}, \tag{4.4b}$$

$$\sum_s \mathbf{B}^{(s)} \left(\mathbf{f}_b^{(s)} - \mathbf{f}_{ext,b}^{(s)} \right) = \mathbf{0}. \tag{4.4c}$$

Here, the local Degree of Freedom (DOF) of subdomain (s) appears separated into internal and boundary (or interface) DOF, denoted $\mathbf{u}_i$ and $\mathbf{u}_b$, respectively. The stiffness matrix and the load vectors are partitioned accordingly. The internal displacements $\mathbf{u}_i^{(s)}$, the boundary displacements $\mathbf{u}_b^{(s)}$, and the boundary loads $\mathbf{f}_b^{(s)}$ are unknowns *a priori*. Eq. (4.4a) represents the local equilibrium of the subdomain, *i.e.*, the discrete version of Eq. (4.2). Eq. (4.4b) and (4.4c) are the discrete versions of the coupling conditions formulated over the cells boundaries, *i.e.*, the discrete versions of Eq. (4.3). Vector $\mathbf{u}_{0,b}$ is a known vector which enforces the Dirichlet boundary conditions. It has often only zero values, as it is the case for homogeneous Dirichlet boundary conditions, for instance. Vector $\mathbf{f}_{ext,b}$ contains the know load vector values coming from the external loads. The coupling operators $\underline{\mathbf{B}}^{(s)}$ and $\mathbf{B}^{(s)}$ can be seen as assembly operators. The primal operator $\mathbf{B}^{(s)}$ is a boolean operator (0 or 1 values only), and the dual operator $\underline{\mathbf{B}}^{(s)}$ is a sign boolean operator (0, -1, or 1 values). The -1 values in $\underline{\mathbf{B}}^{(s)}$ come from the minus sign in Eq. (4.3a). One of the important properties of these coupling operators is their orthogonality:

$$\sum_s \underline{\mathbf{B}}^{(s)} \mathbf{B}^{(s)} = \mathbf{0}. \tag{4.5}$$

The interested reader can find further details on the definition and construction of these coupling operators, for example, in (Gosselet and Rey 2006).

Interface unknowns and iterative solving

The main idea behind DDM consists in solving Eq. (4.4) efficiently. For instance, it is interesting to keep the local equilibriums (i.e., Eq. (4.4a)) separated between the subdomains, such that only these (small) linear systems need to be solved instead of a large global one. More interestingly, such local solves can be naturally computed in parallel. In order to keep this decomposed structure of the problem, one needs to use an iterative process to recover the coupling conditions.

In practice, this can be done as follows. In primal approaches, we define an interface vector of unknowns $\mathbf{u}_B$ which contains all the (global) DOF (displacements and potentially rotations for beam and shell elements) associated with nodes located at the interface. With such a choice, the continuity condition (4.4b) is automatically satisfied, and the iterative process consists in updating $\mathbf{u}_B$ to satisfy the equilibrium of the interface efforts (4.4c). Dual approaches consist in the opposite: an interface effort unknown vector $\boldsymbol{\lambda}_B$ satisfying Eq. (4.4c) is introduced and the iterative process enables to vanish the interface gaps, i.e., verifying Eq. (4.4b). Mathematically, these choices can be formulated as follows:

$$\text{Primal approaches:} \qquad \mathbf{u}_b^{(s)} = \mathbf{B}^{(s)\top}\mathbf{u}_B + \mathbf{u}_{0,b}^{(s)}, \tag{4.6a}$$

$$\text{Dual approaches:} \qquad \mathbf{f}_b^{(s)} = \underline{\mathbf{B}}^{(s)\top}\boldsymbol{\lambda}_B + \mathbf{f}_{ext,b}^{(s)}. \tag{4.6b}$$

Considering the orthogonality condition (4.5), it is rather straightforward to see that the primal and dual approaches will strongly imposed one of the coupling conditions (e.g., by substituting Eq. (4.6a) in Eq. (4.4b)).

In this work, we employ a BDD-like method, which is a primal approach. This method is chosen for its simplicity of use and implementation, as it requires only the resolution of Dirichlet problems on each subdomain. Before going into more details in the following sections, let us summarize and give an overview of the method. Figure 4.6 illustrates the step-by-step procedure of the DDM applied to a 2D lattice structure composed of 2×2 unit cells. The iterative process is described as follows:

1. **Problem definition**: Define the global problem, including the geometry, boundary conditions, and loading conditions.

2. **Domain decomposition**: Divide the global problem into smaller subdomains, in this case, the cells of the lattice structure. At this step, we can build all the

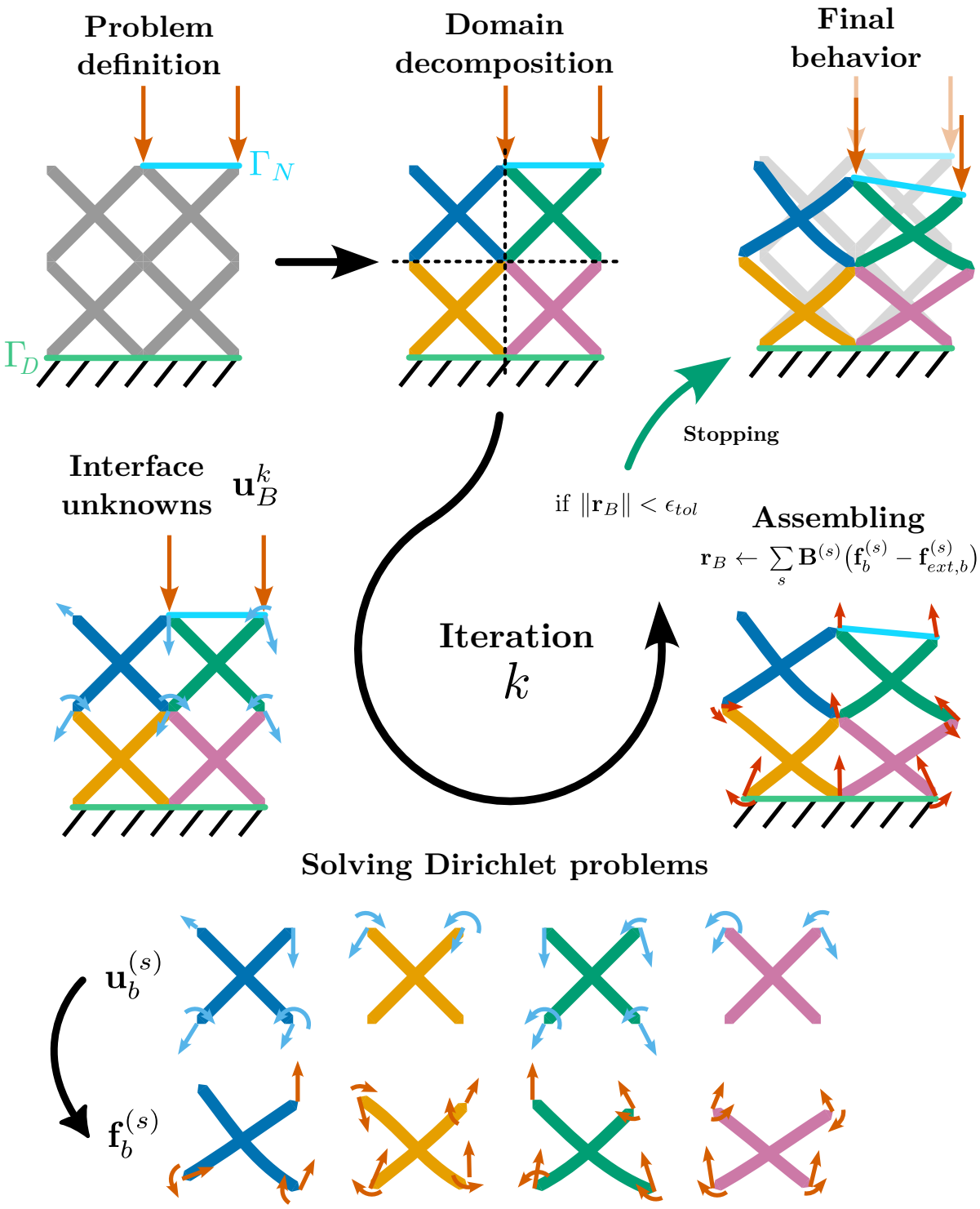

Fig. 4.6 Step-by-step schematic representation of the Balanced Domain Decomposition (BDD) method applied to two-dimensional BCC beam lattice structures. Each node possesses three DOF, corresponding to two translations and one rotation.

following local FE operators:

$$\forall s : \mathbf{K}_{ii}^{(s)}, \mathbf{K}_{ib}^{(s)}, \mathbf{K}_{ib}^{(s)}, \mathbf{K}_{bb}^{(s)},\ \mathbf{u}_{0,b}^{(s)},\ \mathbf{f}_{i}^{(s)}, \mathbf{f}_{ext,b}^{(s)},\ \mathbf{B}^{(s)}.$$

3. **Initializing the interface unknowns**: Give an initial guess for the displacement and rotation DOF associated to interface nodes:

$$\mathbf{u}_B \leftarrow \mathbf{u}_B^0.$$

4. **Solving each subdomain independently**: Evaluate the equilibrium of each subdomain by solving a Dirichlet problem:

$$\begin{aligned}\mathbf{u}_b^{(s)} &\leftarrow \mathbf{B}^{(s)\top}\mathbf{u}_B + \mathbf{u}_{0,b}^{(s)},\\ \mathbf{u}_i^{(s)} &\leftarrow \mathbf{K}_{ii}^{(s)-1}\Big(\mathbf{f}_i^{(s)} - \mathbf{K}_{ib}^{(s)}\mathbf{u}_b^{(s)}\Big).\end{aligned}$$

5. **Assembling reaction forces and moments on the interfaces** : The local reaction forces and moments are evaluated and the equilibrium of the interface loads is tested:

$$\begin{aligned}\mathbf{f}_b^{(s)} &\leftarrow \mathbf{K}_{bi}^{(s)}\mathbf{u}_i^{(s)} + \mathbf{K}_{bb}^{(s)}\mathbf{u}_b^{(s)},\\ \mathbf{r}_B &\leftarrow \sum_s \mathbf{B}^{(s)}\Big(\mathbf{f}_b^{(s)} - \mathbf{f}_{ext,b}^{(s)}\Big).\end{aligned}$$

6. **Updating the interface variables** : If the residual of the interface efforts is small, then the overall solution is achieved. Else, the procedure is repeated: the interface unknowns are updated based on the residual (this step is explained in the next sections):

$$\begin{cases}\text{stop} & \text{if } \|\mathbf{r}_B\| < \epsilon_{tol},\\ \mathbf{u}_B \leftarrow \text{update}(\mathbf{u}_B, \mathbf{r}_B) & \text{else.}\end{cases}$$

4.2.2 Primal interface problem

Local primal Schur complement

As mentioned, in this work we apply a primal domain decomposition approach. In this context, we need to solve local Dirichlet problems: the inputs are boundary displacements (and eventually rotations) $\mathbf{u}_b^{(s)}$ and the needed outputs are the reaction forces (and eventually moments) $\mathbf{f}_b^{(s)}$. The operator that performs this procedure is called the local

primal Schur complement, denoted hereafter $\mathbf{S}_{bb}^{(s)}$. It is given by:

$$\mathbf{S}_{bb}^{(s)} = \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi}^{(s)}\mathbf{K}_{ii}^{(s)-1}\mathbf{K}_{ib}^{(s)}. \tag{4.7}$$

The local Schur complement provides a condensed form of the local equilibrium of a subdomain, expressed in terms of interface quantities:

$$\begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \begin{pmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f}_b^{(s)} \end{pmatrix} \quad \Leftrightarrow \quad \mathbf{S}_{bb}^{(s)}\mathbf{u}_b^{(s)} = \mathbf{f}_b^{(s)}.$$

Global Schur complement and interface problem

If one combines the different equations presented in Section 4.2.1, then an interface problem to be solved appears. With primal approaches, the interface problem is of the form:

$$\mathbf{S}_{BB}\mathbf{u}_B = \mathbf{b}_B, \tag{4.8}$$

where $\mathbf{S}_{BB}$ denotes the global primal Schur complement of the decomposed structure, and $\mathbf{b}_B$ denotes the global condensed right-hand side of the interface problem. Notably, they are made of sums of local contributions:

$$\mathbf{S}_{BB} = \sum_s \mathbf{B}^{(s)}\mathbf{S}_{bb}^{(s)}\mathbf{B}^{(s)\top}, \tag{4.9a}$$

$$\mathbf{b}_B = \sum_s \mathbf{B}^{(s)}\left(\mathbf{f}_{ext,b}^{(s)} - \mathbf{K}_{bi}^{(s)}\mathbf{K}_{ii}^{(s)-1}\mathbf{f}_i^{(s)} - \mathbf{S}_{bb}^{(s)}\mathbf{u}_{0,b}^{(s)}\right). \tag{4.9b}$$

In many case, the local internal load vector $\mathbf{f}_i^{(s)}$ and the local Dirichlet input vector $\mathbf{u}_{0,b}$ are zero vectors. In this case, the global right-hand side $\mathbf{b}_B$ is just the assembly of all local load vectors $\mathbf{f}_{ext,b}^{(s)}$.

Mechanically speaking, the global primal Schur complement $\mathbf{S}_{BB}$ acts similarly to the local primal Schur complement. It takes as an input the interface displacements (and eventually rotations) $\mathbf{u}_B$ and retrieves the associated reaction forces (or the external loads). Evaluating the residual $\mathbf{r}_B = \mathbf{b}_B - \mathbf{S}_{BB}\mathbf{u}_B$ is strictly equivalent to check whereas the coupling condition in terms of equilibrium of the tractions along the subdomain interfaces is satisfied or not (i.e., checking Eq. (4.4c) or even Eq. (4.3b)).

4.2.3 Solving the Interface Problem

Standard domain decomposition approaches

Conjugate gradient method The interface system obtained from the Schur complement is symmetric and positive definite, which makes it particularly well suited for iterative solvers as, for instance, the Conjugate Gradient (CG) method. The CG algorithm is a robust and widely used iterative solver for large linear systems. Originally introduced by Hestenes and Stiefel (1952), the CG method solves linear systems of the form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.10}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive-definite matrix, $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns, and $\mathbf{b} \in \mathbb{R}^n$ is the right-hand side. The CG algorithm iteratively solves the linear system by minimizing the associated quadratic form along a sequence of conjugate directions. At each iteration, such operations are done:

$$\begin{aligned}
&\text{(Current residual)} && \mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k,\\
&\text{(A-conjugate search direction)} && \mathbf{p}^k = \mathbf{r}^k - \mathrm{proj}_{\mathcal{K}^k}(\mathbf{r}^k),\\
&\text{(Optimal step length)} && \alpha^k = \mathrm{argmin}\|\mathbf{b} - \mathbf{A}(\mathbf{x}^k + \alpha\mathbf{p}^k)\|,\\
&\text{(Update the solution)} && \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k\mathbf{p}^k.
\end{aligned}$$

In practice, the method requires at every iteration one single matrix-vector product (i.e., performing $\mathbf{A}\mathbf{p}^k$), and inner products. Efficient implementation of the CG method can be found in most scientific computing libraries, as for instance in SciPy (Virtanen et al. 2020). Using an iterative solver for the DD interface problem (4.8) is crucial, in general, for two main reasons:

1. It enables keeping the decomposed structure of the problem, which in practice enables parallel computing: the application of the global Schur complement is the sum of the application of all local Schur complements.

2. It prevents explicitly building all local Schur complements which would be a rather expensive computational task as this would involve many local solutions (as much as local boundary DOF per subdomain, see Eq. (4.7)). Instead, by using an iterative solver, one only needs to factorize the local sub-matrices $\mathbf{K}_{ii}^{(s)}$, and perform forward-backward substitutions at each iteration.

Preconditioner The convergence rate of the CG method depends on the condition number of the matrix of the system, which may become large in the presence of strong heterogeneities or anisotropic behaviors, as commonly encountered in lattice structures. To mitigate this effect and improve convergence, preconditioners are typically introduced. A preconditioner is a matrix, denoted $\mathbf{M}$, that approximates the inverse of the system matrix $\mathbf{A}$, and is used to transform the original system into a better conditioned one:

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \tag{4.11}$$

The preconditioned system retains the same solution $\mathbf{x}$, but the modified operator $\mathbf{M}^{-1}\mathbf{A}$ has a reduced condition number, leading to faster convergence of the CG algorithm.

The design of an efficient preconditioner is crucial in the context of the DDM, especially when dealing with strongly varying material properties or non-uniform cell geometries. In this particular context, one would also need the preconditioner to take a similar decomposed structure to the interface problem itself. The classical BDD approach makes us of a Neumann preconditioner (Gosselet and Rey 2006; Mandel 1993), which reads as:

$$\tilde{\mathbf{S}}_{BB}^{-1} = \sum_s \mathbf{D}^{(s)}\mathbf{B}^{(s)}\mathbf{S}_{bb}^{(s)+}\mathbf{B}^{(s)\top}\mathbf{D}^{(s)}. \tag{4.12}$$

The diagonal matrices $\mathbf{D}^{(s)}$ are scaled operators that take into account heterogeneities at the interface and enable to better balance the residual between subdomains during the preconditionning (or correction) step. The Neumann preconditioner involves local pure Neumann problems (i.e., the application of the pseudo-inverses $\mathbf{S}_{bb}^{(s)+}$, or in other terms the application of the dual local Schur complements $\mathbf{F}_{bb}^{(s)} = \mathbf{S}_{bb}^{(s)+}$). These dual Schur complements are associated to optimality conditions, which finally leads to form and then solve a coarse problem at every iteration. This coarse problem plays a major role in the scalability of the classical BDD approach. Again, more details might be found in (Gosselet and Rey 2006), for instance.

The specific case of beam-like lattice structure

In the present beam-like lattice setting, the preconditioner from the classical BDD approach is not used. The resulting interface system remains of moderate size, as the number of interface DOF in our cases is relatively small. Beyond this size consideration, all subproblems are somehow similar due to the quasi-periodicity occurring within lattice structures. We will see, in what follows, that this enables to build fast approximation of the local Schur complements $\tilde{\mathbf{S}}_{bb}^{(s)} \approx \mathbf{S}_{bb}^{(s)}$. With these fast local approximations, it is

conceivable to build an approximation of the global Schur complement:

$$\tilde{\mathbf{S}}_{BB} = \sum_s \mathbf{B}^{(s)} \tilde{\mathbf{S}}_{bb}^{(s)} \mathbf{B}^{(s)\top}, \quad \tilde{\mathbf{S}}_{BB} \approx \mathbf{S}_{BB}. \tag{4.13}$$

Then, a sparse LU decomposition (or even Cholesky decomposition) of $\tilde{\mathbf{S}}_{BB}$ can performed and be used to define a preconditioner:

$$\mathbf{M}^{-1} = \mathrm{LU}(\tilde{\mathbf{S}}_{BB})^{-1}. \tag{4.14}$$

Note that this only a valid approach if the computation of the local approximation $\tilde{\mathbf{S}}_{bb}^{(s)}$ is fast, and if the size of the global interface problem is small. Again, it seems doable in our context of truss-based lattice structures modeled with beam elements. In more general settings, with solid elements for instance, one would need to build the preconditioner in a domain decomposition fashion, see for example (Hirschler, Bouclier, et al. 2024).

4.2.4 Application to Beam-like Lattice Structures

The DDM offers several key advantages that make it particularly suitable for the analysis of lattice structures:

- **Problem size reduction**: By decomposing the global problem into smaller local problems, the method reduces the overall dimensionality, making computations more manageable and less memory-intensive. Computing the Schur complement at the subdomain level further condenses the system to the interface DOFs only, which are significantly fewer than the total number of DOFs in the full-scale model.
- **Geometric and functional flexibility**: The method can accommodate a wide range of lattice geometries and loading scenarios, enhancing its adaptability to different design and optimization problems, as mentionned in Section 4.1.
- **Compatibility with heterogeneous physics**: Since each subdomain can be assigned its own material properties and physical conditions, DDM can handle complex geometries and multiphysics behaviors in a consistent and modular framework.
- **Beam-specific advantages**: The method is particularly well-suited for beam-like structures, as the interface DOFs are naturally defined at the nodes where beams connect, simplifying the formulation and implementation. This choice avoids non-conforming mesh issues that typically arise with more complex cell geometries.

- **Parallelizability**: Each subdomain can be solved independently, enabling efficient parallel computation and reducing the overall simulation time.

DDM Implementation Let us first validate the DDM implementation on the three-point bending benchmark case, as previously described in Section 4.1.2. We compare three different approaches:

- **Full-scale FEM**: A full-scale FE simulation of the entire structure using 1D Timoshenko beam elements, which serves as a reference solution for comparison.
- **DDM standard**: A primal DDM in which the subdomain problems are solved by computing each local Schur complement *a priori*, before the iterative solving.
- **DDM periodic**: A primal DDM in which we take advantage of the repetitive nature of the cells to explicitly compute one and only one local Schur complement (as there are all the same), and subsequently assemble the global operator.

The DDM algorithm has been implemented in Python, leveraging the `FEniCSx` (Alnæs et al. 2014; Baratta et al. 2023; Scroggs, Baratta, et al. 2022; Scroggs, Dokken, et al. 2022) library for all FE computations, although the parallel capabilities of the framework are not exploited in the present implementation. The code is available within the open-source framework `pyLattice` (See Appendix A for more details).

Table 4.2 provides a detailed comparison of the three approaches in terms of computational efficiency, accuracy, and other relevant performance indicators. Two types of structures are considered in this study. The first one exhibits periodicity, meaning that all unit cells are identical, while the second one is non-periodic, where each cell is assigned a set of geometric parameters randomly chosen within a limited set of reference geometries. Without a preconditioner, the standard DDM remains eight times more efficient than the full-scale FEM, converging in 465 iterations. This is a bit surprising as all the 250 local Schur complements were computed explicitly. When all the cells are the same as it is the case in the structure type "Periodic" presented in Table 4.2, a preconditioner of the form of Eq. (4.13) can be easily set. Indeed, we can define:

$$\tilde{\mathbf{S}}_{bb}(\boldsymbol{\mu}^{(s)}) = \mathbf{S}_{bb}^{\text{ref}}, \quad \forall s, \tag{4.15}$$

where $\mathbf{S}_{bb}^{\text{ref}}$ is the local Schur complement associated with the (reference) unit cell. This operator is computed only once, leading to a fast construction of the preconditioner. Interestingly, for this periodic lattice structure, such a preconditioner will be *exact* in the

Table 4.2 Numerical experiments illustrating the accuracy and computational efficiency of the different simulation approaches.

Method	Structure type	Precon-ditioner	Comp. time (s)	Relative error	No. of unknowns	No. of iter.
Full-scale FEM	Periodic	—	506.2	Reference	981,294	—
Full-scale FEM	Random	—	512.3	Reference	981,294	—
DDM standard	Periodic	✗	58.8	$\approx 10^{-6}$	13,592	465
DDM standard	Periodic	✓	14.5	$\approx 10^{-11}$	13,592	1
DDM standard	Random	✗	665.4	$\approx 10^{-6}$	13,592	5702
DDM standard	Random	✓	47.3	$\approx 10^{-6}$	13,592	275
DDM standard wrong penal.	Random	✓	59.5	$\approx 10^{-3}$	13,592	209
DDM periodic	Periodic	✗	49.2	$\approx 10^{-6}$	13,592	466
DDM periodic	Periodic	✓	5.2	$\approx 10^{-11}$	13,592	1
DDM periodic	Random	—	—	—	—	—

sense that we have:

$$\mathbf{M}^{-1} = \mathbf{S}_{BB}^{-1}. \tag{4.16}$$

It corresponds exactly to the inverse of the global Schur complement. As a result, the iterative process converges in a single iteration as seen in Table 4.2, since the interface problem is solved directly, as detailed in Section 4.2.3. In this case, a 35-fold improvement is achieved (analysis time is 14.5 s versus the $\approx$ 500 s for the direct FEM solve), while the relative error is reduced to machine precision. However, for non-periodic structures, the computational time without a preconditioner exceeds that of the full-scale FEM, due to the large number of iterations required for convergence. By constructing a preconditioner based on a single reference Schur complement, the results show a significant reduction in the number of iterations.

Remark. Particular attention must be paid to the analysis of non-periodic structures combined with the beam penalization method introduced in Chapter 3. The results show a reduced agreement with the full-scale FEM, not because of the CG solver convergence, but because the problem being solved is not strictly equivalent to the reference case. In the DDM preprocessing stage, each Schur complement is computed independently, without considering the influence of neighboring cells. As a result, the boundary beams are inaccurately evaluated. To better reflect the mechanical behavior of cells within a lattice structure, periodic boundary mapping is applied to compute the penalization of each beam more accurately, as shown in Figure 4.7(a). Applying a periodic boundary mapping implies accounting for the beams located on the opposite sides of the cell when computing the penalization factor. However, this introduces a difference with the full-scale FEM, which computes the penalization directly on the exact geometry, as illustrated in Figure 4.7(b). The same issue occurs for the corner beams of the structure, as demonstrated in Figure 4.7(c). Therefore, a periodic boundary mapping must also be applied at the structure scale.

When all cells in the structure are identical, the analysis via the DDM can be further improved. The preprocessing phase becomes even simpler. One single Schur complement is computed as all local problems are equal:

$$\mathbf{S}_{bb}(\boldsymbol{\mu}^{(1)}) = \mathbf{S}_{bb}(\boldsymbol{\mu}^{(2)}) = \cdots = \mathbf{S}_{bb}(\boldsymbol{\mu}^{(N_s)}).$$

In other words, what was done for the preconditioning step in Eq. (4.15) can be done for the interface problem itself. Instead of precomputing independently the local operators per sub-domains, one can do it only once. This further decreases the overall computational time and leads to speedups of nearly two orders of magnitude.

Overall, the DDM approach proves to be computationally efficient for lattice structure simulations, but only when the formulation is properly defined. The DDM periodic approach yields highly promising results, but in its current form, it cannot be applied to non-periodic or random structures. The main bottleneck still lies in the efficient computation of the Schur complement for each cell configuration. By exploiting the geometric and mechanical similarity between cells, a promising strategy consists in employing Reduced Order Model (ROM) techniques to efficiently construct the Schur complements across the parameter space. In other words, we are willing to exploit the fact that:

$$\mathbf{S}_{bb}(\boldsymbol{\mu}^{(1)}) \sim \mathbf{S}_{bb}(\boldsymbol{\mu}^{(2)}) \sim \cdots \sim \mathbf{S}_{bb}(\boldsymbol{\mu}^{(N_s)}),$$

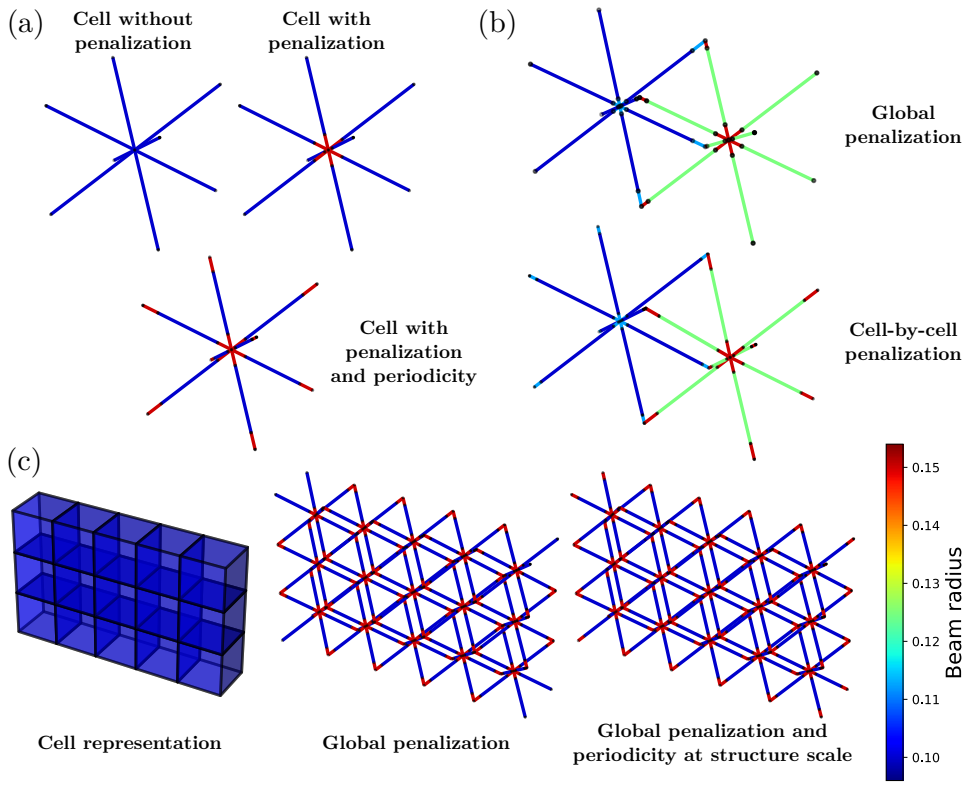

Fig. 4.7 Illustration of the penalization method applied in different configurations and showing the effect of periodicity. (a) Comparison at the cell level, (b) Difference between global and cell-by-cell penalization, and (c) Illustration with a $5 \times 1 \times 3$ BCC lattice structure. The color scale indicates the beam radius.

in order to build the DD interface problem in a cost effective manner. We will see hereafter that such approaches enable efficient and scalable evaluations without the need for repeated high-fidelity simulations, thereby significantly reducing the overall computational cost.

4.3 Surrogate Model for Fast Analysis

Surrogate models are powerful tools that enable efficient approximations of complex systems by interpolating or extrapolating their behavior from a limited number of high-fidelity simulations. A wide range of surrogate modeling techniques has been developed, spanning various levels of complexity and accuracy, depending on the amount of available data and the required precision. However, the most critical aspect of these methods lies in the careful selection of both the variable to approximate and the construction of an

appropriate training dataset, as these choices strongly influence the overall performance of the surrogate model.

A fairly straightforward strategy in the present context consists in constructing a surrogate model for the shape-parameterized local Schur complements. More specifically, in this section, we make use the Reduced Basis (RB) method to construct projection-based ROM in combination with hyperreduction techniques. In other terms, we seek for approximations of the following form:

$$\mathbf{S}_{bb}(\boldsymbol{\mu}) \approx \sum_{i=1}^{N} \alpha_i(\boldsymbol{\mu}) \mathbf{S}_{bb}^i, \tag{4.17}$$

where $\{\mathbf{S}_{bb}^i\}_{i=1}^N$ defines the reduced basis, and $\alpha_i(\boldsymbol{\mu})$ are interpolation weights that depend on the current geometrical parameters $\boldsymbol{\mu}$. Having an affine approximation of the local Schur complement as given in Eq. (4.17) can be computationally very efficient if the size N of the reduced basis is small (compared to the size of the internal DOF), and if the evaluation of the interpolation functions α_i is fast itself. Therefore, the challenges are twofold: how to build a small yet representative reduced basis and how to quickly project in the generated reduced space.

Remark. Hereafter, we choose to skip the subscripts $(\cdot)_{bb}$ in the local Schur complements. This notation was introduced in Section 4.2 to differentiate these local operators to the global Schur complement $\mathbf{S}_{BB}$, where B denoted global boundary DOF whereas b denoted local ones. Thus, in what follows, $\mathbf{S}$ denotes the local Schur complement.

4.3.1 Reduced Basis via Greedy Algorithm

Reduced-Order Modeling Principles

ROM aims to approximate high-fidelity numerical simulations by projecting the solution $u(\boldsymbol{\mu}) \in \mathcal{V}_h$ onto a low-dimensional subspace, where $\boldsymbol{\mu} \in \mathcal{P}$ is the parameter vector and $\mathcal{V}_h$ denotes the high-dimensional FE space (Hesthaven et al. 2016). Such methods are particularly relevant in parameterized problems where the full-order solution, typically associated with high computational cost, must be evaluated for a wide range of parameter values, as encountered in optimization or real-time applications. A reduced basis $\{\varphi_i\}_{i=1}^N \subset \mathcal{V}_h$ is constructed such that the reduced-order approximation of the solution reads:

$$u_N(\boldsymbol{\mu}) = \sum_{i=1}^{N} \alpha_i(\boldsymbol{\mu})\, \varphi_i, \tag{4.18}$$

Algorithm 1: Greedy algorithm for reduced basis construction

Input: Training set $\mathcal{P}$, tolerance ε
Output: Reduced basis $\mathcal{V}_n$

Initialize the reduced basis $\mathcal{V}_1 = \{u(\boldsymbol{\mu}^{(1)})\}$;
for $n = 2$ **to** $N_{\max}$ **do**
  foreach $\boldsymbol{\mu} \in \mathcal{P}$ **do**
    Evaluate error estimator $\Delta(\boldsymbol{\mu})$ between high-fidelity solution $u(\boldsymbol{\mu})$ and its reduced approximation $u_N(\boldsymbol{\mu}) \in \mathcal{V}_{n-1}$;
  Select parameter with maximum error: $\boldsymbol{\mu}^{(n)} = \arg\max_{\boldsymbol{\mu}} \Delta(\boldsymbol{\mu})$;
  if $\Delta(\boldsymbol{\mu}^{(n)}) < \varepsilon$ **then**
    stop;
  Compute new snapshot $u(\boldsymbol{\mu}^{(n)})$;
  Enrich reduced space: $\mathcal{V}_n = \mathcal{V}_{n-1} \cup \{u(\boldsymbol{\mu}^{(n)})\}$;

with $N \ll \dim(\mathcal{V}_h)$. The reduced space is typically generated from a set of snapshots $\{u(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_s}$ obtained from high-fidelity simulations at selected parameter values.

Greedy Algorithm

Among the different strategies for building the reduced space, the greedy algorithm offers an efficient and systematic approach. The greedy method is adopted due to its straightforward implementation and because its reduced basis elements are directly drawn from the global solution space, unlike POD or SVD-based techniques, thereby facilitating the preservation of essential model properties. The greedy algorithm iteratively enriches the reduced basis by selecting, at each step, the parameter $\boldsymbol{\mu}$ that maximizes an error indicator over a predefined training set. The corresponding high-fidelity solution $u(\boldsymbol{\mu})$ is then added to the reduced space.

The greedy procedure is presented in Algorithm 1. It involves a user-defined tolerance ε that controls the termination of the greedy algorithm. It defines the maximum admissible error for the reduced-order approximation over the training set. The error indicator $\Delta(\boldsymbol{\mu})$ corresponds to the maximum distance between a given high fidelity data and its projection onto the current reduced space. The greedy algorithm provides several benefits in comparison to the commonly adopted Proper Orthogonal Decomposition (POD) method. Specifically, the reduced basis is built based on the selection of several representative snapshots (and not based on linear combinations of all snapshots as for POD modes). In other words, it identifies key parameter values denoted hereafter $\boldsymbol{\mu}_{\text{ref}}^{i}$ which capture the essential variations of the underlying quantity.

Application to local DD operators

The construction of a reduced basis in the case of the local Schur complements is done as follows. Let us first consider a training sample set $\mathcal{P}^n = \{\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(n)}\}$, and generate the associated snapshots (i.e., we compute the Schur complement for each training point). In practice, the local Schur complements are dense symmetric matrices. These matrices can be vectorized, and one can form a snapshot matrix S:

$$\mathsf{S} = \left[\mathrm{vec}(\mathbf{S}(\boldsymbol{\mu}^{(1)})), \dots, \mathrm{vec}(\mathbf{S}(\boldsymbol{\mu}^{(n)}))\right], \quad \mathsf{S} \in \mathbb{R}^{M \times n}, \tag{4.19}$$

where the number of row is $M = n_b^2$ or $M = n_b(n_b-1)/2$ depending if one uses the fact that the local Schur complements are symmetric matrices of size $n_b \times n_b$ or not.

Applying a greedy procedure to the snapshot matrix will generate a reduced basis $\mathsf{S}_{\mathrm{ref}}$ based on the selection of N_{ref} reference training points:

$$\mathsf{S}_{\mathrm{ref}} = \left[\mathrm{vec}(\mathbf{S}(\boldsymbol{\mu}_{\mathrm{ref}}^{1})), \dots, \mathrm{vec}(\mathbf{S}(\boldsymbol{\mu}_{\mathrm{ref}}^{N_{\mathrm{ref}}}))\right], \quad \mathsf{S}_{\mathrm{ref}} \in \mathbb{R}^{M \times N_{\mathrm{ref}}}. \tag{4.20}$$

This reduced basis spans a reduced space which should be suitable for the approximation of the Schur complement. This approximation can be done via orthogonal projection:

$$\mathbf{S}(\boldsymbol{\mu}) \approx \mathsf{S}_{\mathrm{ref}} \boldsymbol{\alpha}(\boldsymbol{\mu}), \tag{4.21a}$$

$$\boldsymbol{\alpha}(\boldsymbol{\mu}) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{N_{\mathrm{ref}}}}{\arg\min} \|\mathbf{S}(\boldsymbol{\mu}) - \mathsf{S}_{\mathrm{ref}} \boldsymbol{\beta}\|_2^2 = (\mathsf{S}_{\mathrm{ref}}^\top \mathsf{S}_{\mathrm{ref}})^{-1} \mathsf{S}_{\mathrm{ref}}^\top \mathbf{S}(\boldsymbol{\mu}). \tag{4.21b}$$

Such a projection requires to solve an $N_{\mathrm{ref}} \times N_{\mathrm{ref}}$ linear system. Often, the reduced basis $\mathsf{S}_{\mathrm{ref}}$ might be built as an orthonormal matrix during the construction (even in the greedy algorithm) such that this step is straightforward and given by:

$$\boldsymbol{\alpha}(\boldsymbol{\mu}) = \mathsf{S}_{\mathrm{ref}}^\top \mathbf{S}(\boldsymbol{\mu}). \tag{4.22}$$

Remark. In Eq. (4.21)-(4.22) we have made an abuse of notation by skipping the vectorization of the Schur complement for the sake of clarity (we directly use $\mathbf{S}$ instead of $\mathrm{vec}(\mathbf{S})$. This abuse of notation will be often employed hereafter.

Numerical investigation

Let us apply the greedy reduction strategy to the shape-parametrized unit cells as introduced in Section 4.1.1. During the greedy procedure (see Algo. 1), the error

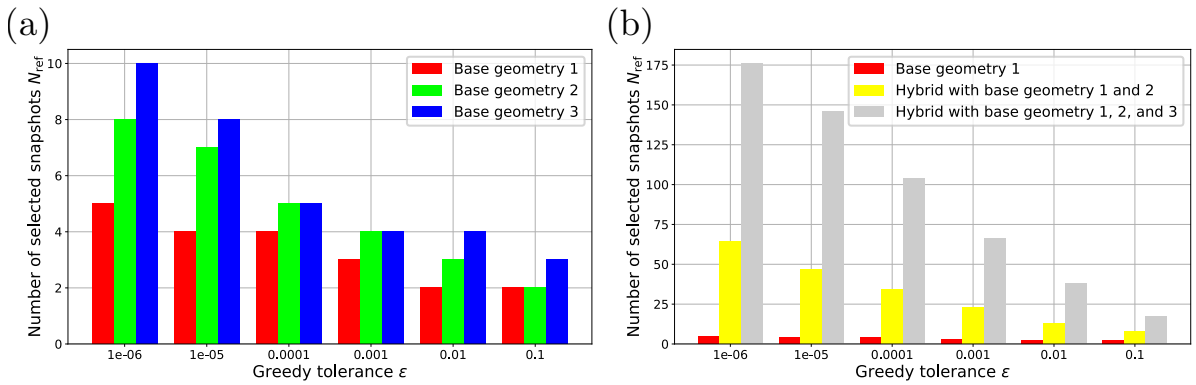

Fig. 4.8 Snapshot reduction dependency with the error tolerance $\varepsilon_{\text{greedy}}$ using the greedy algorithm, for both (a) base unit cell geometries and (b) hybrid configurations.

indicator used here is the norm of the linear least squares residual:

$$\begin{aligned}\Delta_S(\boldsymbol{\mu}) &= \min_{\boldsymbol{\alpha}} \|\mathbf{S}(\boldsymbol{\mu}) - \mathsf{S}_{\text{ref}}\boldsymbol{\alpha}\|_2, \\ &= \|\mathbf{S}(\boldsymbol{\mu}) - \mathsf{S}_{\text{ref}}(\mathsf{S}_{\text{ref}}^\top \mathsf{S}_{\text{ref}})^{-1}\mathsf{S}_{\text{ref}}^\top \mathbf{S}(\boldsymbol{\mu})\|_2.\end{aligned} \tag{4.23}$$

In this study, the Schur complements are evaluated on uniformly sampled reference configurations of the parameter space, using a step size of 0.01. Figure 4.8 shows the size of the reduced bases required to reach a prescribed accuracy for different base geometries, as well as hybrid configurations. The results demonstrate that the number of selected snapshots depends not only on the desired precision but also on the underlying base geometry used (See Figure 4.8(a)). For instance, for target precision of $\varepsilon_{\text{greedy}} = 10^{-6}$, the number of selected snapshots is 5 for the geometry 1, 8 for the geometry 2, and 10 for the geometry 3. This variation in the number of snapshots reflects the diversity of the mechanical behavior of each base unit cell.

A second observation is that the number of selected snapshots (i.e., the size of the reduced basis) required to achieve a given precision increases with the number of shape parameters, as shown in Fig. 4.8(b). Again, for a precision of $\varepsilon_{\text{greedy}} = 10^{-6}$, the RB size rises from 5 for a unit cell composed of a single base geometry, to 64 when two base geometries are combined, and up to 176 when three base geometries are considered. This increase in the RB size is expected, as the complexity of the mechanical behavior increases with the number of shape parameters. It leads to a larger design space and requires more reference configurations to accurately capture the variations of the Schur complement. This increase is often referred to as the curse of dimensionality: the use of linear RB is limited in practice to design space of low dimension. In our case, we limit the number of shape parameters defined at the unit cell level to three.

In this thesis, different design spaces are employed primarily for simplicity of representation; however, all proposed methods are fully compatible with any of the considered design spaces.

4.3.2 A variety of Surrogate Modeling Techniques

Having a reduced basis S_{ref} is not sufficient to have a fast and accurate surrogate of the local Schur complement. Indeed, the projection in Eq. 4.21 is not usable in practice as it involves the computation of the full Schur complement in order to get the reduced coefficients $\boldsymbol{\alpha}(\boldsymbol{\mu})$. Therefore, additionnaly to the construction of the reduced basis, projection-based reduced-order models additionally involve hyper-reduction or interpolation techniques to get a fast (approximated) evaluation of these coefficients (Bhattacharyya et al. 2025). In other words, we are looking for surrogate models of the Schur complement of the form:

$$\tilde{\mathbf{S}}(\boldsymbol{\mu}) = \mathsf{S}_{\text{ref}}\tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu}), \tag{4.24}$$

where the vector valued function $\tilde{\boldsymbol{\alpha}} : \mathcal{P} \to \mathbb{R}^{N_{\text{ref}}}$ is fast to evaluate and a good approximation of the initial function $\boldsymbol{\alpha}$.

Specifically, a wide range of hyper-reduction and interpolation techniques can be employed to approximate the projection function α_i over the parameter space, including:

- **Nearest neighbor**: to approximate the function by selecting the closest interpolation point.
- **Linear interpolation**: to approximate the function by interpolating between the two nearest interpolation points in the parameter space.
- **Polynomial, spline, or radial basis interpolation**: to approximate the solution using smoother interpolation schemes, at the expense of higher computational cost.
- **Kriging, Gaussian process regression**: to model the function as a realization of a Gaussian process, providing uncertainty quantification along with the predictions.
- **Neural networks**: to approximate the function by learning nonlinear relationships between the parameters and the Schur complement, with increased computational requirements.

In practice, the mentioned approaches require snapshots of the map to be approximated. In our case, it consists in generating an interpolation sample set $\mathcal{P}_{\text{intrp}} = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^{N_{\text{intrp}}}\}$

of dimension N_{intrp}, and evaluating the reduced coefficients for any given sample point:

$$\boldsymbol{\alpha}(\boldsymbol{\mu}^i) = \boldsymbol{\alpha}^i_{\text{intrp}}, \quad i = 1, \ldots, N_{\text{intrp}}. \tag{4.25}$$

The choice of an interpolation (or approximation) method depends on the desired trade-off between computational cost and accuracy. In what follows, we explore different interpolation strategies and compare their performance in terms of accuracy and computational efficiency, starting with the simplest methods and progressively moving towards more complex ones.

Nearest neighbor

The simplest approach is to select the coefficients of the closest training point $\boldsymbol{\mu}^k \in \mathcal{P}_{\text{intrp}}$ to the current parameter $\boldsymbol{\mu} \in \mathcal{P}$, measured in the Euclidean distance:

$$\tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu}) = \boldsymbol{\alpha}^k_{\text{intrp}}, \quad \text{where} \quad k = \underset{i=1,\ldots,N_{\text{intrp}}}{\arg\min} \|\boldsymbol{\mu} - \boldsymbol{\mu}^i\|_2^2. \tag{4.26}$$

Although this strategy is extremely fast, it yields a piecewise-constant surrogate $\tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu})$, resulting in discontinuities that can be detrimental to gradient-based optimization later on (see, in particular, Chapter 5).

Linear barycentric interpolation

A more refined approach is to interpolate the coefficients within the convex simplex $\mathcal{T}$ formed by interpolation configurations surrounding $\boldsymbol{\mu}$. If $\{\boldsymbol{\mu}^k_{\text{vert}} \in \mathcal{P}_{\text{intrp}}\}_{k=1}^{p+1}$ is the set of vertices of $\mathcal{T}$, the coefficients are reconstructed as:

$$\tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu}) = \sum_{k=1}^{p+1} w_k \, \boldsymbol{\alpha}(\boldsymbol{\mu}^k_{\text{vert}}), \quad \text{with} \quad \sum_{k=1}^{p+1} w_k = 1, \; w_k \geq 0, \tag{4.27}$$

where the weights w_k are the barycentric coordinates of $\boldsymbol{\mu}$ in the simplex $\mathcal{T}$. These are computed by solving:

$$\boldsymbol{V}\,\boldsymbol{\lambda} = \boldsymbol{\mu} - \boldsymbol{\mu}^{p+1}_{\text{vert}}, \qquad \boldsymbol{V} = \begin{bmatrix} \boldsymbol{\mu}^1_{\text{vert}} - \boldsymbol{\mu}^{p+1}_{\text{vert}} & \cdots & \boldsymbol{\mu}^p_{\text{vert}} - \boldsymbol{\mu}^{p+1}_{\text{vert}} \end{bmatrix}, \tag{4.28}$$

then setting $w_k = \lambda_k$ for $k = 1, \ldots, p$ and $w_{p+1} = 1 - \sum_{k=1}^{p} \lambda_k$. Both nearest-neighbor and barycentric interpolation are simple to implement and computationally efficient, but their accuracy strongly depends on the density and distribution of the interpolation points within the parameter space.

Radial Basis Functions Interpolation

The smoothness and regularity of a surrogate model directly determine its suitability for gradient-based optimization methods. Typical examples include polynomial interpolation, splines, and radial RBF interpolation. Polynomial interpolation builds a single polynomial passing through all reference points. Its strengths are simplicity, smoothness, and analytical clarity. However, it quickly faces problems: oscillations for many points, global sensitivity where each new point changes the whole curve, and poor scalability in higher dimensions. For this reason, polynomials, while elegant, are unreliable for complex or irregular datasets.

RBF interpolation offers a more flexible alternative by expressing the surrogate as a weighted sum of radially symmetric functions centered at each reference point. It handles scattered data well and can achieve high smoothness. Its main strengths lie in being mesh-free, applicable in any dimension without a grid, and adaptable enough to capture both smooth and complex behaviors. On the other hand, it remains costly because dense systems scale poorly as the number of points grows, it can suffer from ill-conditioning depending on the shape parameter, and each new point still influences the entire interpolant. In this work, we adopt RBF interpolation to approximate for the reduced coefficients $\boldsymbol{\alpha}(\boldsymbol{\mu})$, ensuring a smooth representation across the parameter space. Mathematically, the evaluation of a coefficient α_i at a parameter $\boldsymbol{\mu}$ is approximated by:

$$\tilde{\alpha}_i(\boldsymbol{\mu}) = \sum_{j=1}^{N_{\text{intrp}}} w_{i,j}\, \varphi\Big(\|\boldsymbol{\mu} - \boldsymbol{\mu}^j\|\Big)\,, \tag{4.29}$$

where φ is a chosen radial basis kernel (e.g., Gaussian, multiquadric, thin-plate spline), and the weights $w_{i,j}$ are obtained by enforcing the interpolation condition:

$$\tilde{\alpha}_i(\boldsymbol{\mu}^j) = \alpha^j_{\text{interp},i}, \quad j = 1, \ldots, N_{\text{intrp}}. \tag{4.30}$$

In this work, the thin-plate spline kernel is employed, defined as:

$$\varphi(r) = r^2 \log r, \quad \varphi(0) = 0, \tag{4.31}$$

which provides a smooth (C^∞ away from $r = 0$ or C^1 everywhere) and well-suited basis for capturing the smooth variation of the reduced coefficients across the parameter space.

Neural Network-Based Learning of Reduced Coefficients

Machine learning techniques, and in particular Neural Network (NN)s, offer a powerful framework for approximating highly nonlinear mappings between input and output variables. In the increasingly complex design spaces of modern engineering applications, NNs have emerged as a promising alternative to traditional surrogate modeling techniques. An advantage of NN is their ability to capture complex dependencies difficult to model with classical interpolation methods. Moreover, using differentiable activation functions, NNs can provide smooth and continuous approximations of the reduced coefficients, ensuring that the surrogate model is amenable to gradient-based optimization techniques. Nevertheless, the use of NN also comes with notable drawbacks. Training a network typically requires a large dataset of reference solutions, which can be computationally expensive to generate in the offline phase. The training process itself involves hyperparameter tuning, risk of overfitting, and may lead to models whose internal representations are difficult to interpret. In addition, unlike standard interpolants, NNs do not guarantee exact reproduction of the training data, and their performance strongly depends on the quality and diversity of the dataset. These aspects must therefore be carefully addressed when designing a NN-based surrogate model.

Let us consider here a simple fully-connected feedforward NN to learn the map $\boldsymbol{\alpha} : \boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p \mapsto \boldsymbol{\alpha}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\text{ref}}}$. Two different strategies can be employed. The first strategy consists in training a separate network for each scalar function α_i, since the functions are independent from each other. This approach is simpler and provides greater flexibility in selecting the architecture and hyperparameters of each network. However, it becomes less efficient when the number of coefficients is large, as it requires training and maintaining multiple networks. The second strategy, which is adopted in this work, involves training a single network to learn the vector-valued function at once. More specifically, we define a dense NN composed of d hidden layers $\{\boldsymbol{l}_i\}_{i=1,\dots,d}$. Each layer $\boldsymbol{l}_i$ consists of affine transformations (defined by weights $\mathbf{W}_i$ and bias $\mathbf{b}_i$) composed with an activation (vector-valued) function $\boldsymbol{a}_i$. We denote the set of parameters (all weights and bias) by $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$, and are looking for approximation of the reduced function $\boldsymbol{\alpha}$ of the following form:

$$
\begin{aligned}
\tilde{\boldsymbol{\alpha}}_\theta^{\text{NN}}(\boldsymbol{\mu}) &= (\boldsymbol{l}_d \circ \boldsymbol{l}_{d-1} \circ \ldots \circ \boldsymbol{l}_1)(\boldsymbol{\theta})(\boldsymbol{\mu}), \\
&= \underbrace{\boldsymbol{a}_d\Big(\mathbf{W}_d \cdot \underbrace{\boldsymbol{a}_{d-1}\Big(\mathbf{W}_{d-1} \cdot \ldots \cdot \underbrace{\boldsymbol{a}_1(\mathbf{W}_1 \cdot \boldsymbol{\mu} + \mathbf{b}_1)}_{\text{output of hidden layer } \boldsymbol{l}_1} + \ldots + \mathbf{b}_{d-1}\Big)}_{\text{output of hidden layer } \boldsymbol{l}_{d-1}} + \mathbf{b}_d\Big)}_{\text{output of final layer } \boldsymbol{l}_d}.
\end{aligned}
\tag{4.32}
$$

The parameters $\boldsymbol{\theta}$ are identified during a so-called training phase, which consists in solving an optimization problem. The most standard approach is to minimize the residual between the network predictions and the observations, through the use of the mean square error:

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^{N_\theta}}{\text{minimize}}\ \mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}), \quad \text{where} \quad \mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{intrp}}} \sum_{\boldsymbol{\mu}^i\in\mathcal{P}_{\text{intrp}}} \|\tilde{\boldsymbol{\alpha}}_\theta^{\text{NN}}(\boldsymbol{\mu}^i) - \boldsymbol{\alpha}_{\text{intrp}}^i\|_2^2. \tag{4.33}$$

The training phase is done via gradient-based algorithms. This phase is far from a trivial task in general (Glorot and Bengio 2010), due to the non-convexity of the loss function, the high number of the parameters, etc.

4.3.3 Model Behavior and Accuracy Assessment

Performance Analysis of the NN Architectures

The NN model requires careful adjustment of the training parameters to guarantee numerical stability and reliable convergence. A first example is the dataset, which must be sufficiently large and diverse to ensure proper training and generalization compared to interpolation methods. To this end, the number of evaluation data points is increased by one order of magnitude (from 10 to 100), and the parameter range is extended to $[0.005, 0.105]$ to mitigate boundary effects. The adopted single-network approach requires careful consideration of the different scales present in the coefficients $\boldsymbol{\alpha}$. Each coefficient is independently normalized through min–max scaling, ensuring comparable magnitudes across all components and facilitating stable network training. The architecture of the network also plays a crucial role in the performance of the surrogate model, as it depends on the complexity and the number of functions to be learned. Several NN architectures were evaluated by varying both the number of hidden layers and the number of neurons per layer. Table 4.3 summarizes the NN architectures tested for learning the reduced coefficients $\tilde{\alpha}_i(\boldsymbol{\mu})$. The error metric is defined as a global relative error over all samples and all coefficients:

$$\varepsilon_\alpha = \sqrt{\sum_{i=1}^{N_{\text{test}}} \left\|\boldsymbol{\alpha}(\boldsymbol{\mu}^i) - \tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu}^i)\right\|_2^2} \Bigg/ \sqrt{\sum_{i=1}^{N_{\text{test}}} \left\|\boldsymbol{\alpha}(\boldsymbol{\mu}^i)\right\|_2^2}, \tag{4.34}$$

which corresponds to the normalized L^2-error aggregated over the entire dataset. The results show that deeper and wider networks do not necessarily yield better accuracy, highlighting a trade-off between model complexity and generalization ability. Further

Table 4.3 Summary of tested NN architectures and corresponding errors. The best-performing architectures for each case are highlighted in orange.

Case #1: 1 geometries, $\varepsilon_{\text{greedy}} = 10^{-3}$						
Input	Layer 1	Layer 2	Layer 3	Layer 4	Output	Relative error ε_α
1	32	32	–	–	3	8.189×10^{-4}
Case #2: 1 geometries, $\varepsilon_{\text{greedy}} = 10^{-6}$						
Input	Layer 1	Layer 2	Layer 3	Layer 4	Output	Relative error ε_α
1	64	64	–	–	5	1.349×10^{-3}
Case #3: 3 geometries, $\varepsilon_{\text{greedy}} = 10^{-3}$						
Input	Layer 1	Layer 2	Layer 3	Layer 4	Output	Relative error ε_α
3	128	128	–	–	66	4.351×10^{-3}
3	512	512	–	–	66	4.551×10^{-3}
3	64	64	64	–	66	7.315×10^{-3}
3	128	128	128	–	66	2.631×10^{-3}
3	256	256	256	–	66	3.429×10^{-3}
3	512	512	512	–	66	3.526×10^{-3}
3	32	64	128	–	66	3.210×10^{-3}
3	128	256	128	–	66	2.955×10^{-3}
3	256	256	128	–	66	3.518×10^{-3}
3	128	128	128	128	66	2.735×10^{-3}
3	64	128	256	128	66	3.147×10^{-3}
Case #4: 3 geometries, $\varepsilon_{\text{greedy}} = 10^{-6}$						
Input	Layer 1	Layer 2	Layer 3	Layer 4	Output	Relative error ε_α
3	256	256	256	–	176	7.109×10^{-3}
3	512	512	512	–	176	4.774×10^{-3}

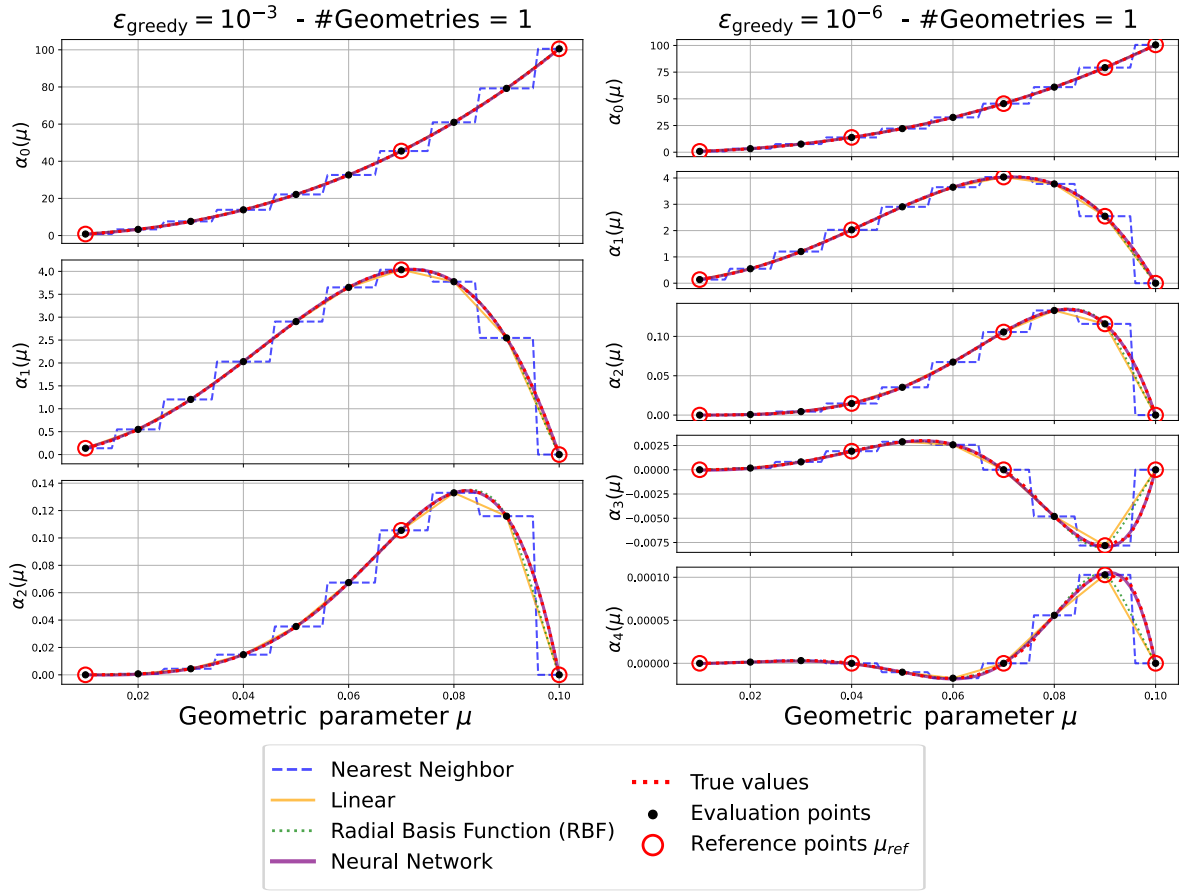

Fig. 4.9 Comparison of the surrogate model predictions for a single geometry with two different greedy precision $\varepsilon_{\text{greedy}}$ (Left: $\varepsilon_{\text{greedy}} = 10^{-3}$, Right: $\varepsilon_{\text{greedy}} = 10^{-6}$).

investigation is required to reduce the error and identify more efficient architectures and hyperparameters.

Performance Comparison of the Surrogate Models

After constructing each surrogate model, it is crucial to assess its accuracy and behavior across the parameter space. Figure 4.9 illustrates the evolution of the reduced coefficients $\alpha_i(\mu)$ for a single geometry, obtained with two different greedy tolerances $\varepsilon_{\text{greedy}}$. The true coefficients display smooth variations with respect to the parameter for the most part, although some irregularities are observed, particularly for the fourth and fifth coefficients when using a greedy tolerance of $\varepsilon_{\text{greedy}} = 10^{-6}$. After further investigation, these irregularities were attributed to the computation of penalized junctions, although their exact origin remains unclear. While such irregularities may affect the accuracy of the surrogate models, their overall influence remains limited and does not significantly alter

the general trends of the coefficients. The nearest-neighbor model exhibits a piecewise-constant behavior, resulting in a discontinuous approximation (not C^0). In contrast, the linear interpolation model yields a piecewise-affine function, which is continuous (C^0) but not differentiable ($\notin C^1$), thus providing a smoother approximation of the coefficients across the parameter space. However, this limited regularity is not sufficient for gradient-based optimization, since the absence of continuous derivatives prevents the consistent computation of sensitivities with respect to the parameters. The RBF surrogates demonstrate good approximation capabilities for moderately varying functions, but their accuracy decreases for coefficients exhibiting strong variations, as observed for the fourth and fifth modes with a greedy tolerance of $\varepsilon_{\text{greedy}} = 10^{-6}$. Their smoothness depends on the selected basis function. In this work, thin-plate splines are employed, which provide C^1 continuity. This level of regularity is sufficient to enable gradient-based optimization, as it guarantees the existence of continuous first derivatives with respect to the parameters. NN-based surrogates exhibit the highest approximation capabilities, accurately capturing the variations of all coefficients across the parameter space. Their smoothness depends on the selected activation function. In the present work, the `tanh` activation is employed, which yields C^∞ models and therefore ensures continuously differentiable coefficients and gradients suitable for gradient-based optimization.

Table 4.4 Global relative error ε_α of different surrogate models for varying numbers of geometries and $\varepsilon_{\text{greedy}}$.

No. Geometries	$\boldsymbol{\varepsilon_{\text{greedy}}}$	**Nearest Neighbor**	**Linear**	**RBF**	**NN**
1	10^{-3}	7.985×10^{-2}	4.969×10^{-3}	2.953×10^{-3}	8.189×10^{-4}
1	10^{-6}	7.985×10^{-2}	4.969×10^{-3}	2.953×10^{-3}	1.349×10^{-3}
3	10^{-3}	9.955×10^{-2}	6.313×10^{-3}	1.336×10^{-3}	2.631×10^{-3}
3	10^{-6}	9.955×10^{-2}	6.313×10^{-3}	1.336×10^{-3}	4.774×10^{-3}

Table 4.4 reports the relative errors obtained with the different surrogate models. The performance of each surrogate model was evaluated using the error metric defined in Equation (4.34). The first observation is that the greedy tolerance $\varepsilon_{\text{greedy}}$ does not affect the accuracy of the nearest-neighbor, linear, and RBF models. This behavior is expected, as these models do not involve a training phase; their accuracy is mainly determined by the density and distribution of the reference configurations. However, choosing a greedy tolerance $\varepsilon_{\text{greedy}}$ smaller than the intrinsic precision of the surrogate model is not beneficial, since it does not further improve the overall predictive accuracy. The total

error can be decomposed into its principal contributions as follows:

$$\|\mathbf{S}(\boldsymbol{\mu}) - \tilde{\mathbf{S}}(\boldsymbol{\mu})\| \leq \underbrace{\|\mathbf{S}(\boldsymbol{\mu}) - \mathsf{S}_{\text{ref}}\boldsymbol{\alpha}(\boldsymbol{\mu})\|}_{\text{projection error}} + \|\mathsf{S}_{\text{ref}}\| \underbrace{\|\boldsymbol{\alpha}(\boldsymbol{\mu}) - \tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu})\|}_{\text{interpolant error}}, \tag{4.35}$$

where $\mathbf{S}(\boldsymbol{\mu})$ denotes the exact Schur complement, $\mathsf{S}_{\text{ref}}\boldsymbol{\alpha}(\boldsymbol{\mu})$ its projection onto the reduced basis, and $\tilde{\mathbf{S}}(\boldsymbol{\mu})$ the surrogate approximation. The error is therefore decomposed into two parts: the projection error and the surrogate model error. The first term on the right-hand side of equation (4.35) represents the projection error, which is controlled by the greedy tolerance $\varepsilon_{\text{greedy}}$. The second term corresponds to the surrogate model error, which depends on the chosen surrogate modeling technique. The overall error is therefore governed by the combined effects of the greedy tolerance and the surrogate model. When one of these two contributions largely exceeds the other, it becomes the dominant source of error.

Table 4.5 Evolution of surrogate model errors as a function of the number of snapshots used (single geometry, $\varepsilon_{\text{greedy}} = 10^{-6}$).

No. Snapshots	**Nearest Neighbor**	**Linear**	**RBF**
10	7.985×10^{-2}	4.969×10^{-2}	2.953×10^{-3}
22	3.860×10^{-2}	1.246×10^{-3}	6.478×10^{-5}
53	1.932×10^{-2}	1.935×10^{-4}	1.010×10^{-5}
80	1.416×10^{-2}	1.399×10^{-4}	5.001×10^{-6}

Effect of Dataset Size on Accuracy

In order to improve the accuracy of the surrogate model, different snapshot dataset sizes were tested for a single geometry with $\varepsilon_{\text{greedy}} = 10^{-6}$, as reported in Table 4.5. The RBF surrogate model exhibits a significantly greater improvement in accuracy than the other surrogates, with the error decreasing from 2.953×10^{-3} for 10 snapshots to 5.001×10^{-6} for 80 snapshots. Nevertheless, the achieved accuracy remains slightly above the greedy tolerance of 10^{-6}, indicating that further refinement or alternative surrogate strategies may be required to fully exploit the potential of the reduced basis framework.

The second observation is that the accuracy of the surrogate model improves with its level of complexity. The nearest-neighbor model consistently produces the largest errors, while linear interpolation provides moderate improvements. The RBF surrogates achieve even higher accuracy. In contrast, the NN-based models exhibit a different

behavior, achieving the lowest errors for single-geometry cases, but slightly higher errors for hybrid configurations composed of three base geometries. This can be explained by the increased complexity of the functions to be learned when the number of geometries grows. One factor that limits the accuracy of NN-based models is that the predictions at the reference points are not guaranteed to be exact, unlike interpolation-based methods. As a result, the performance of NN-based surrogates tends to be more uniform across the parameter space, whereas interpolation-based methods achieve perfect accuracy at the reference points but exhibit larger errors in between. As previously discussed, multiple architectures were tested, but reaching higher predictive accuracy remains challenging. A more systematic tuning of the hyperparameters might further enhance the performance, but this aspect was left for future work. One possible way to improve the accuracy of NN-based surrogates is to train a separate network for each coefficient α_i, allowing the architecture and training strategy to be tailored to the complexity of the corresponding function.

Related Works on NN–ROM Accuracy

Several recent works have combined reduced-order modeling with NN, reporting relative errors in the same order of magnitude as those obtained in this study. For instance, Q. Wang et al. (2019) report mean relative errors of 1.28% for a global POD–NN and 0.67% for a local POD–NN on unsteady combustion cases, with corresponding POD projection errors of 0.083% and 0.095%, highlighting the gap between projection and learned regression of coefficients. In the POD–Deep Learning (DL)–ROM framework, Fresca and Manzoni (2022) present several nonlinear PDEs cases with error indicators ranging from 1.40×10^{-2} to 1.63×10^{-3} depending on the physics considered. In their earlier DL–ROM study on latent dynamics, Fresca, Dede', et al. (2021) reported an accuracy of $\varepsilon_{\text{rel}} = 5.87 \times 10^{-3}$, using a more complex architecture for comparison with the present work. More recently, Tannous et al. (2024) proposed an ML–based ROM for transient nonlinear structural mechanics, reporting an average displacement relative error of about 0.7% across 15 validation cases, illustrating a similar 10^{-2} regime for field-level accuracy. Overall, these results confirm that NN enhanced reduced-order models typically achieve relative errors in the range 10^{-3}–10^{-2}, which is consistent with the performance observed in the present work.

Performance Comparison on Benchmark Problems

To assess the performance of the different surrogate models in practical scenarios, three benchmark problems introduced in Section 4.1.2 are considered. Test points are generated

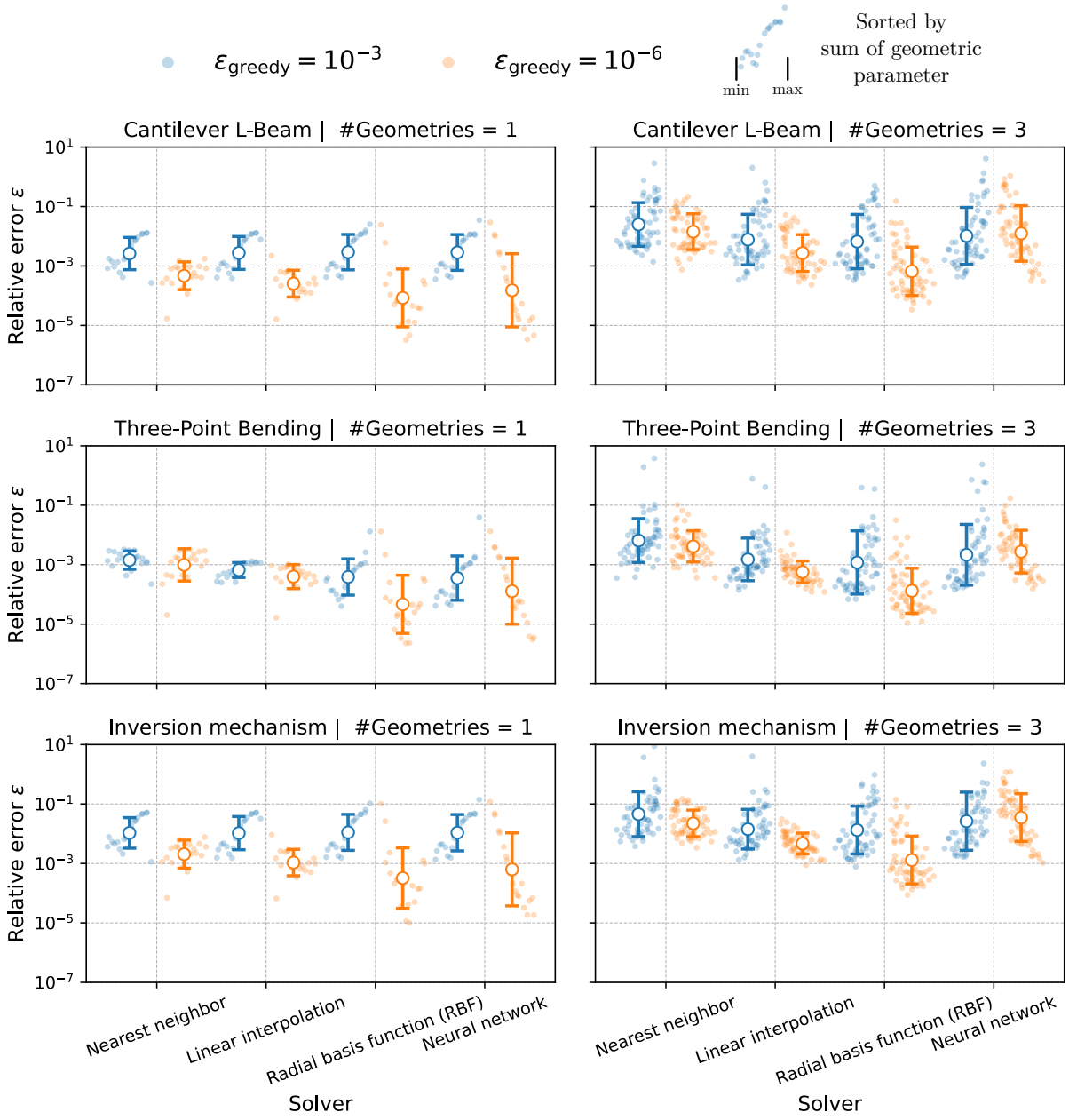

Fig. 4.10 Relative error distributions on benchmark problems for different surrogate models, compared against FEM simulations. Each dots correspond to individual test cases, sorted by increasing sum of the geometrical parameters. White markers denote the mean relative error, and error bars indicate one standard deviation.

using LHS, with 20 samples for the single-geometry case and 60 samples for the three-geometry case. Figure 4.10 reports the relative errors computed over the simulation unknowns, namely the displacements and rotations at the boundary nodes, compared to the reference FEM results. The first observation illustrates the error decomposition

introduced in Equation 4.35. In the three-geometry case, the nearest-neighbor surrogate indicates that the surrogate error is predominant, as the mean error remains nearly constant around 10^{-2} for both $\varepsilon_{\text{greedy}} = 10^{-3}$ and $\varepsilon_{\text{greedy}} = 10^{-6}$, implying that the surrogate error largely surpasses the projection error. As a result, for larger values of $\varepsilon_{\text{greedy}}$, the accuracy of the surrogate is less important since the total error is already driven by the projection error. The nearest-neighbor and linear surrogates yield higher mean errors but smaller standard deviations, whereas the RBF and NN models achieve lower mean errors at the expense of larger variability across test cases. This behavior can be explained by the fact that nearest-neighbor and linear interpolations are less accurate but more stable, while RBF and NN surrogates are generally more accurate but sensitive to local variations in the coefficients. In particular, NN-based models produce relatively uniform errors on the coefficients $\alpha_i(\boldsymbol{\mu})$, which, when combined with the smaller contributions of the Schur complement at low geometric parameter values, leads to a higher overall error. This trend is clearly illustrated in the single-geometry case with $\varepsilon_{\text{greedy}} = 10^{-6}$, where the NN-based surrogate fails to reduce the overall error despite its smoother coefficient approximations. However, larger geometric parameter values are better predicted, resulting in lower errors in these cases. One last observation is that the relative error is dependant of the problem to solve. Errors are lower by an order of magnitude between the inversion mechanism and the three-point bending problem. This behavior can be attributed to the number of cells in the structure, which is higher in the three-point bending case, leading to smoother global responses.

4.4 Embedding Physical Knowledge into Neural Network

In the present case, linear elasticity, small deformations, and quasi-static conditions are assumed. Under these assumptions, the mechanical response of the structure remains relatively simple, and the Schur complement depends only on the geometric parameters of the cell $\boldsymbol{\mu}$. However, if dynamic effects are considered, the Schur complement may become time-dependent as:

$$\mathbf{M}(\boldsymbol{\mu})\ddot{\mathbf{u}}(t) + \mathbf{C}(\boldsymbol{\mu})\dot{\mathbf{u}}(t) + \mathbf{K}(\boldsymbol{\mu})\mathbf{u}(t) = \mathbf{f}_{\text{ext}}(t), \tag{4.36}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ the damping matrix, $\mathbf{K}$ the stiffness matrix, and $\mathbf{f}_{\text{ext}}$ the external force vector. In the case of harmonic analysis, the Schur complement will

depend both on the frequency ω and the geometrical parameters $\boldsymbol{\mu}$, as:

$$\mathbf{D}(\omega, \boldsymbol{\mu}) = \mathbf{K}(\boldsymbol{\mu}) + i\omega\mathbf{C}(\boldsymbol{\mu}) - \omega^2\mathbf{M}(\boldsymbol{\mu}). \tag{4.37}$$

Constructing a surrogate model by means of a linear method as the projection-based ROM used in Section 4.3.2 becomes significantly more challenging in such general dynamic and eventually non-linear settings, as the parameter space must now account for additional dependencies on time or frequency. The size of the reduced basis is likely to grow significantly, making the overall procedure ineffective.

To avoid this issue of dimensional catastrophe, non-linear ROM gains large interest in the machine learning community (Brivio et al. 2024; Daniel et al. 2020; Hess et al. 2021), as it might capture features more efficiently. The following developed strategy falls into this category of methods. Furthermore, we aim to embed physical knowledge into the surrogate formulation to reduce the number of data needed to train the surrogate. This section introduces a PANN-type of surrogate model relying on an energy formulation.

4.4.1 Embedding Physics for Efficient and Generalizable Surrogate Models

Motivations

The general objective is to express the equilibrium of each cell by means of a surrogate model capable of predicting its mechanical response for a given set of geometrical parameters and loading conditions. In other words, we are looking for a surrogate model that approximates the interface behavior at the cell level. What is needed during the iterative solving, within primal domain decomposition methods, is the computation of the external loads associated to imposed displacements at the cell boundary (see Section 4.2). Thus, the non-linear map to be approximated is of the form:

$$\mathbf{f}_b : (\mathbf{u}_b; \boldsymbol{\mu}) \in \mathbb{R}^{n_b} \times \mathcal{P} \mapsto \mathbf{f}_b(\mathbf{u}_b; \boldsymbol{\mu}) \in \mathbb{R}^{n_b}, \tag{4.38}$$

where $\mathbf{u}_b$ denotes the vector of boundary DOF (translations and rotations) of size n_b, and $\boldsymbol{\mu}$ represents the geometric parameters associated to the cell. The output of the map $\mathbf{f}_b(\mathbf{u}_b; \boldsymbol{\mu})$ is a vector of size n_b too, that contains the nodal forces and moments.

In the case of the hybrid unit cell described in Section 4.1.1, the cell has 26 boundary nodes. Each node possesses six DOF (three translational and three rotational), resulting in a total of $n_b = 156$ DOF. The map $\mathbf{f}_b$ takes inputs in a space of dimension $n_b + N_p = 159$ and map them in a output space of dimension $n_b = 156$. Training a feedforward NN with

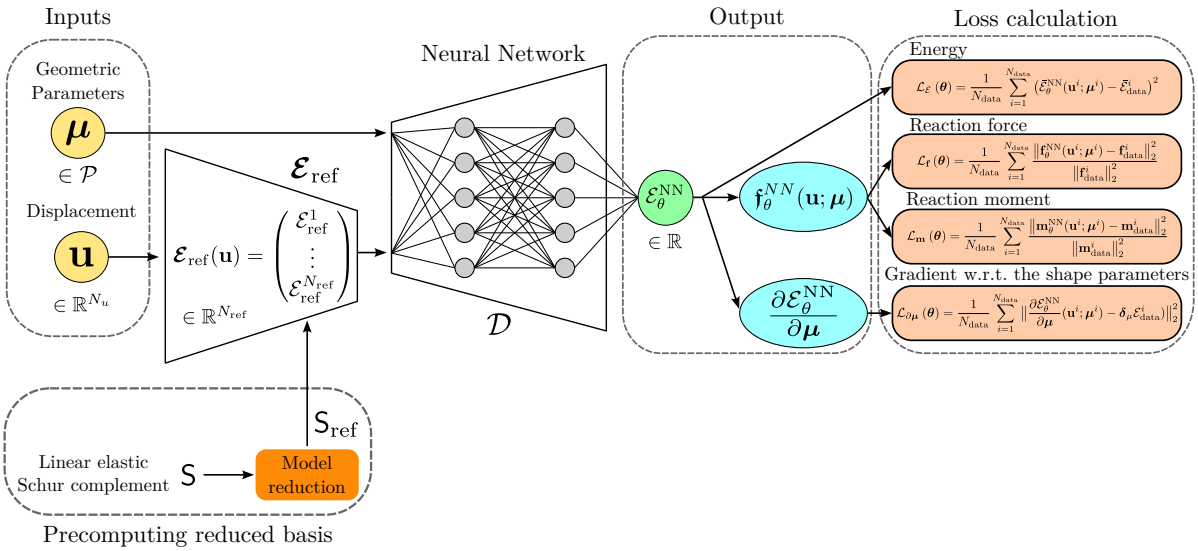

Fig. 4.11 Proposed PANN architecture.

such a high-dimensional input-output spaces requires a deep architecture and a large dataset, which in practice is not trackable (and might lead to high computational cost).

A more effective strategy might be to exploit the physical knowledge of the problem to reduce the dimensionality and complexity of the learning task. To this end, the proposed surrogate model relies on a PANN that integrates physical constraints directly into the network. Compared to well-established Physics-Informed Neural Network (PINN)s, which incorporate physical laws as additional terms in the loss function, a PANN goes further by embedding physical principles directly within the network architecture and its input–output representation. Such approaches often improve both the efficiency of the training process and the generalization capabilities of the model. The architecture of the developed PANN is built upon an energy-based formulation, where the network predicts the strain energy of the cell instead of the full set of reaction forces and moments. This choice significantly reduces the output space to a single scalar quantity, while still allowing the reaction loads to be recovered through differentiation. The overall architecture is illustrated in Figure 4.11.

Remark. For the sake of clarity, we will drop the subscript b in what follows. This means that vector $\mathbf{u}$ will not denote the whole local DOF vector, but only the boundary ones (and same holds true for the load vector).

Energy-Based Approach

A key physical insight that simplifies the surrogate formulation are to use energy principles. If the material is elastic (possibly hyperelastic, large strains, and buckling), the

Castigliano's-type relation holds true, i.e., the applied load is the derivative of strain energy with respect to displacement:

$$\mathfrak{f} = \frac{\partial \mathcal{E}}{\partial \mathbf{u}}, \tag{4.39}$$

where $\mathcal{E}$ denotes the strain energy (of a cell).

As a result, the entire mechanical response at the interface can be derived from the energy potential. Therefore, instead of *surrogating* directly the map of interest $\mathfrak{f}$ (see Eq. (4.38)), we can rather consider the energy map:

$$\mathcal{E} : (\mathbf{u}; \boldsymbol{\mu}) \in \mathbb{R}^{N_u} \times \mathcal{P} \mapsto \mathcal{E}(\mathbf{u}; \boldsymbol{\mu}) \in \mathbb{R}. \tag{4.40}$$

In order to recover the applied load as need by the DDM, one can use the energy principle from Eq. (4.39). In other words, the load vector is obtained by evaluating the partial derivative of the strain energy w.r.t. the (boundary) displacement. Interestingly, such derivatives can be efficiently computed when using surrogate models. This is true, for instance, in the case of NN models thanks to AD Margossian 2019. This substantial reduction in the dimensionality of the map to be learned often simplifies the training task and might improve the robustness of the model.

Explicit latent space

The second key insight is the reformulation of the input space to further simplify training. Instead of relying solely on displacements and geometrical parameters as inputs, we incorporate precomputed strain energies, which we refer to as reference (strain) energies $\mathcal{E}^i_{\text{ref}}, i = 1, \dots, N_{\text{ref}}$. In this work, we define these reference energies from precomputed (linear elastic) primal local Schur complement corresponding to well-chosen shape parameters $\boldsymbol{\mu}_{\text{ref}}$:

$$\mathcal{E}^i_{\text{ref}} : \mathbf{u} \in \mathbb{R}^{N_u} \mapsto \mathcal{E}^i_{\text{ref}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^\top \mathbf{S}(\boldsymbol{\mu}^i_{\text{ref}})\mathbf{u} \in \mathbb{R}, \quad i = 1, \dots, N_{\text{ref}}. \tag{4.41}$$

Let us also denote by $\boldsymbol{\mathcal{E}}_{\text{ref}}$ the vector-value function defined by:

$$\boldsymbol{\mathcal{E}}_{\text{ref}} : \mathbf{u} \in \mathbb{R}^{N_u} \mapsto \boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}) = \begin{pmatrix} \mathcal{E}^1_{\text{ref}} & \dots & \mathcal{E}^{N_{\text{ref}}}_{\text{ref}} \end{pmatrix}^\top \in \mathbb{R}^{N_{\text{ref}}}. \tag{4.42}$$

In the context of machine learning, the dimensionality reduction (from $\mathbb{R}^{N_u}$ to $\mathbb{R}^{N_{\text{ref}}}$) is refered to as encoding, the function $\boldsymbol{\mathcal{E}}_{\text{ref}}$ as the encoder, and the vector $\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u})$ as the latent variables (or latent representation, latent vector, etc.). The strain energy of a

shape-parameterized unit cell $\mathcal{E}(\mathbf{u};\boldsymbol{\mu})$ will then be written as a function of this latent variables:

$$\mathcal{E}(\mathbf{u};\boldsymbol{\mu}) = \mathcal{D}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}), \tag{4.43}$$

where the function $\mathcal{D}$ is of the following form:

$$\mathcal{D} : (\mathbf{e};\boldsymbol{\mu}) \in \mathbb{R}^{N_{\text{ref}}} \times \mathcal{P} \mapsto \mathcal{D}(\mathbf{e};\boldsymbol{\mu}) \in \mathbb{R}, \tag{4.44}$$

and can be seen as the decoder.

The motivations for writing the strain energy as in Eq. (4.43) are multiple. Interestingly, we can link this formulation with the projection based ROM approach described in Section 4.3, especially the surrogate models formulated as in Eq. (4.24). This would be strictly equivalent to defining the function $\mathcal{D}$ as:

$$\mathcal{D}^{\text{lin}}(\mathbf{e};\boldsymbol{\mu}) = \mathbf{e} \cdot \tilde{\boldsymbol{\alpha}}(\boldsymbol{\mu}). \tag{4.45}$$

Instead of such a linear approach (the strain energy of a shape-parameterized cell is written as a linear combination of the strain energy of several reference cells), we are looking for a more advanced and possibly non-linear formulation of $\mathcal{D}$. The hope is the size of N_{ref} (i.e., the number of modes or latent variables) can be chosen small even for larger numbers of the shape parameters N_p. In other words, we hope that the use of such a non-linear reduced order modeling technique will prevent the dimensional catastrophe. The initial motivation was also to extend the use of ROM to non-linear mechanical behavior such as hyperelastic material, for instance. In this case, the strain energy is no longer quadratic w.r.t. the displacement, and the Schur complement depends on the displacements, making linear ROM inefficient (due to too large reduced basis). Non-linear mechanical behavior will not be investigate in what follows, but the developed approach was done in that perspective.

The explicit construction of the encoder as in Eq. (4.42) enables to automatically fulfill some interesting mechanical properties. It naturally acts as a filter that removes rigid body modes from the input vector $\mathbf{u}$. Indeed, if we denote by $\boldsymbol{\mathcal{U}}_{\text{RBMs}} \subset \mathbb{R}^{N_u}$ the (discrete) space of rigid body motions, we have:

$$\forall \mathbf{u}^0 \in \boldsymbol{\mathcal{U}}_{\text{RBMs}} : \boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}^0) = \mathbf{0}, \quad \text{and} \quad \boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u} + \mathbf{u}^0) = \boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}). \tag{4.46}$$

Zero energy modes are thus easily trackable as there are naturally associated with zero-valued latent vector. Furthermore, the encoder $\boldsymbol{\mathcal{E}}_{\text{ref}}$ preserves the symmetry of the shape parameterized unit cell. Let us define by $\mathbf{u}_{\text{trac x}}$ and $\mathbf{u}_{\text{trac y}}$ a boundary displacement vector

that impose a tensile stretch to the cell in direction x and y, respectively. These two input vectors will be associated to the same latent vector, i.e. $\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}_{\text{trac x}}) = \boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}_{\text{trac y}})$. Note that this is only true since our unit cells have some symmetries, see Section 4.1.1. As a result, these two similar cases will likely lead to the same strain energy, by construction.

Furthermore, the chosen encoder enables to preserve, naturally, the balance of applied forces and moments. Indeed, by substituting Eq. (4.43) into Eq. (4.39), one can see that the applied loads are expressed as:

$$\mathfrak{f}(\mathbf{u};\boldsymbol{\mu}) = \frac{\partial \mathcal{E}}{\partial \mathbf{u}}(\mathbf{u};\boldsymbol{\mu}) = \left[\frac{\mathrm{d}\boldsymbol{\mathcal{E}}_{\text{ref}}}{\mathrm{d}\mathbf{u}}(\mathbf{u})\right]^{\top} \frac{\partial \mathcal{D}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}). \tag{4.47}$$

The Jacobian matrix $\mathrm{d}\boldsymbol{\mathcal{E}}_{\text{ref}}/\mathrm{d}\mathbf{u} \in \mathbb{R}^{N_{\text{ref}} \times N_u}$ reads as:

$$\frac{\mathrm{d}\boldsymbol{\mathcal{E}}_{\text{ref}}}{\mathrm{d}\mathbf{u}}(\mathbf{u}) = \begin{bmatrix} \mathbf{S}(\boldsymbol{\mu}^1_{\text{ref}})\mathbf{u} & \mathbf{S}(\boldsymbol{\mu}^2_{\text{ref}})\mathbf{u} & \cdots & \mathbf{S}(\boldsymbol{\mu}^{N_{\text{ref}}}_{\text{ref}})\mathbf{u} \end{bmatrix}^{\top}. \tag{4.48}$$

Let us define a basis $\mathbf{R} \in \mathbb{R}^{N_u \times 6}$ that spans the space of rigid motions, i.e., $\boldsymbol{\mathcal{U}}_{\text{RBMs}} = \text{span}(\mathbf{R})$. The following relationship holds true:

$$\mathbf{S}(\boldsymbol{\mu}^i_{\text{ref}})\mathbf{R} = \mathbf{0}, \quad i = 1, \dots, N_{\text{ref}}. \tag{4.49}$$

Interestingly, the same basis $\mathbf{R}$ enables to write the optimality condition, or in other term express the balance of forces and moments. A load vector $\mathfrak{f}$ is balanced if the following condition is satisfied:

$$\mathbf{R}^{\top}\mathfrak{f} = \mathbf{0}. \tag{4.50}$$

One can see that this condition is automatically fulfilled, by construction, when the load vector is expressed as in Eq. (4.47):

$$\begin{aligned}
\mathbf{R}^{\top}\mathfrak{f}(\mathbf{u};\boldsymbol{\mu}) &= \mathbf{R}^{\top}\left[\frac{\mathrm{d}\boldsymbol{\mathcal{E}}_{\text{ref}}}{\mathrm{d}\mathbf{u}}(\mathbf{u})\right]^{\top} \frac{\partial \mathcal{D}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}) \\
&= \mathbf{R}^{\top}\begin{bmatrix} \mathbf{S}(\boldsymbol{\mu}^1_{\text{ref}})\mathbf{u} & \mathbf{S}(\boldsymbol{\mu}^2_{\text{ref}})\mathbf{u} & \cdots & \mathbf{S}(\boldsymbol{\mu}^{N_{\text{ref}}}_{\text{ref}})\mathbf{u} \end{bmatrix} \frac{\partial \mathcal{D}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}) \\
&= \begin{bmatrix} \cancelto{0}{\mathbf{R}^{\top}\mathbf{S}(\boldsymbol{\mu}^1_{\text{ref}})\mathbf{u}} & \cancelto{0}{\mathbf{R}^{\top}\mathbf{S}(\boldsymbol{\mu}^2_{\text{ref}})\mathbf{u}} & \cdots & \cancelto{0}{\mathbf{R}^{\top}\mathbf{S}(\boldsymbol{\mu}^{N_{\text{ref}}}_{\text{ref}})\mathbf{u}} \end{bmatrix} \frac{\partial \mathcal{D}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}) \\
&= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \frac{\partial \mathcal{D}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u});\boldsymbol{\mu}) \\
&= \mathbf{0}, \quad \forall (\mathbf{u};\boldsymbol{\mu}) \in \mathbb{R}^{N_u} \times \mathcal{P}.
\end{aligned}$$

Therefore, by construction, the balance of forces and moments is strongly satisfied for any input $(\mathbf{u};\boldsymbol{\mu})$. Lastly, evaluating the load vector via Eq. (4.47) for any rigid body motions returns a zero vector by construction:

$$\mathbf{f}(\mathbf{u};\boldsymbol{\mu}) = \mathbf{0}, \quad \forall (\mathbf{u};\boldsymbol{\mu}) \in \boldsymbol{\mathcal{U}}_{\text{RBMs}} \times \mathcal{P}. \tag{4.51}$$

The proof is very similar to the one showing that the balance of forces and moments is fulfilled, as the Jacobian matrix is null for any $\mathbf{u} \in \boldsymbol{\mathcal{U}}_{\text{RBMs}}$.

By structuring the model around strain energy, we ensure that fundamental physical principles, such as symmetries, balance of forces and moments, zero energy associated to rigid body motions, etc., are intrinsically respected. This approach not only enhances the physical consistency of the predictions but also eliminates the need for additional constraints or corrections during training. As a result, the model naturally adheres to the governing mechanical laws, leading to a more generalizable surrogate for structural analysis.

The developed Physics-Augmented Neural Network

The non-linear ROM is obtained by introducing a feedforward NN to represent the mapping $\mathcal{D}$, which was named the decoder, see Eq. (4.44). Let us detail its formulation. We introduce an approximated map, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$, of the following form:

$$\begin{aligned} \mathcal{D}^{\text{NN}}_\theta : \ & \mathbb{R}^{N_{\text{ref}}} \times \mathcal{P} \to \mathbb{R} \\ & (\mathbf{e};\boldsymbol{\mu}) \ \mapsto (\boldsymbol{l}_{\text{out}} \circ \boldsymbol{l}_{d-1} \circ \ldots \circ \boldsymbol{l}_2 \circ \boldsymbol{l}_{\text{in}})(\boldsymbol{\theta})(\mathbf{e};\boldsymbol{\mu}), \end{aligned} \tag{4.52}$$

where the d-dense layers are given by:

- First (input) layer with n_1 neurons:

$$\begin{aligned} \boldsymbol{l}_{\text{in}}(\boldsymbol{\theta}) : \ & \mathbb{R}^{N_{\text{ref}}} \times \mathcal{P} \to \mathbb{R}^{n_1} \\ & (\mathbf{e};\boldsymbol{\mu}) \ \mapsto \boldsymbol{a}_1(\mathbf{W}_1^{\mu} \cdot \boldsymbol{\mu} + \mathbf{W}_1^{\mathcal{E}} \cdot \mathbf{e} + \mathbf{b}_1) \end{aligned} \tag{4.53}$$

  with weights $\mathbf{W}_1^{\mu} \in \mathbb{R}^{n_1 \times N_p}$ and $\mathbf{W}_1^{\mathcal{E}} \in \mathbb{R}^{n_1 \times N_{\text{ref}}}$, bias $\mathbf{b}_1 \in \mathbb{R}^{n_1}$, and an activation function $\boldsymbol{a}_1 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1}$.

- Several hidden layers with n_i neurons:

$$\boldsymbol{l}_i(\boldsymbol{\theta}) : \mathbf{v} \in \mathbb{R}^{n_{i-1}} \ \mapsto \ \boldsymbol{a}_i(\mathbf{W}_i \cdot \mathbf{v} + \mathbf{b}_i) \in \mathbb{R}^{n_i}, \quad i = 2, \ldots, d-1, \tag{4.54}$$

with weights $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$, bias $\mathbf{b}_i \in \mathbb{R}^{n_i}$, and an activation function $\boldsymbol{a}_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$.

- Last (output) layer with one single neuron:

$$\boldsymbol{l}_{\text{out}}(\boldsymbol{\theta}) : \mathbf{v} \in \mathbb{R}^{n_{d-1}} \;\mapsto\; a_{\text{out}}(\mathbf{w}_d \cdot \mathbf{v} + b_d) \in \mathbb{R}, \tag{4.55}$$

with weights $\mathbf{w}_d \in \mathbb{R}^{n_{d-1}}$, bias $b_d \in \mathbb{R}$, and an activation function $a_{\text{out}} : \mathbb{R} \to \mathbb{R}$.

The total number of training parameters $\boldsymbol{\theta}$ is:

$$N_\theta = n_1(N_p + N_{\text{ref}} + 1) + \sum_{l=2}^{d-1} n_l(n_{l-1} + 1) + n_{d-1} + 1.$$

If the size of all layers is the same (except the output layer), the number of training parameters is then:

$$N_\theta = (d-2)n^2 + n(N_p + N_{\text{ref}} + d) + 1, \quad d \geq 2.$$

By substituting the trainable decoder $\mathcal{D}_\theta^{\text{NN}}$ into Eq. (4.43) and (4.47), the strain energy and the applied load are, respectively, expressed as:

$$\mathcal{E}_\theta^{\text{NN}}(\mathbf{u}; \boldsymbol{\mu}) = \mathcal{D}_\theta^{\text{NN}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}); \boldsymbol{\mu}), \tag{4.56}$$

$$\mathfrak{f}_\theta^{\text{NN}}(\mathbf{u}; \boldsymbol{\mu}) = \left[\mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^1)\mathbf{u} \quad \cdots \quad \mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^{N_{\text{ref}}})\mathbf{u}\right] \frac{\partial \mathcal{D}_\theta^{\text{NN}}}{\partial \mathbf{e}}(\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}); \boldsymbol{\mu}) \tag{4.57}$$

The partial derivative $\partial \mathcal{D}_\theta^{\text{NN}} / \partial \mathbf{e}$ can be efficiently computed thanks to AD (basically a long chain rule of differentiation along all layers of the NN).

Remark. We could eventually compute the tangent operator (i.e., the Schur complement) by taking the second derivatives of the strain energy. This would read as:

$$\begin{aligned}\mathbf{S}_\theta^{\text{NN}} = &\left[\mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^1) \quad \dots \quad \mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^{N_{\text{ref}}})\right] \cdot \frac{\partial \mathcal{D}_\theta^{\text{NN}}}{\partial \mathbf{e}} \\ &+ \left[\mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^1)\mathbf{u} \quad \cdots \quad \mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^{N_{\text{ref}}})\mathbf{u}\right] \frac{\partial^2 \mathcal{D}_\theta^{\text{NN}}}{\partial \mathbf{e}^2} \left[\mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^1)\mathbf{u} \quad \cdots \quad \mathbf{S}(\boldsymbol{\mu}_{\text{ref}}^{N_{\text{ref}}})\mathbf{u}\right]^\top.\end{aligned}$$

Such a computation is not done in what follows.

4.4.2 Training Strategy for the PANN

Normalization Strategies to Stabilize Learning

Architecture of the NN is important for learning the most efficient mapping between inputs and outputs. However, equally crucial is the preprocessing of the data, particularly normalization. Normalization plays a central role in stabilizing the learning process and improving convergence, particularly because most activation functions (e.g., `tanh`, `sigmoid`) are highly sensitive to input scaling. Unnormalized data can push the input values into the saturation zones of these functions, where gradients vanish and learning becomes inefficient. By constraining the input data within an appropriate range, normalization ensures that the activation functions operate in their most responsive region, thus facilitating efficient gradient propagation. For the proposed PANN, the `tanh` activation function is employed, producing outputs within the interval $[-1, 1]$. Consequently, a normalization strategy is adopted to rescale the input and output data to this range.

More specifically, in practice we define the feedforward NN on normalized data, which here consists in considering the following trainable map:

$$\bar{\mathcal{D}}_\theta^{\mathrm{NN}} : (\bar{\mathbf{e}}; \bar{\boldsymbol{\mu}}) \in [-1,\ 1]^{N_{\mathrm{ref}}} \times [-1,\ 1]^{N_p} \mapsto \bar{\mathcal{D}}_\theta^{\mathrm{NN}}(\bar{\mathbf{e}}; \bar{\boldsymbol{\mu}}) \in [-1,\ 1]. \tag{4.58}$$

Since the strain energy spans several orders of magnitude, we first apply a logarithmic transformation to compress its range, thereby preventing small values from being overshadowed by larger ones. To ensure numerical stability during training, the logarithmic transformation is subsequently combined with an affine min–max normalization onto the interval $[-1,\ 1]$. In other word, we define the energy normalization function $\mathcal{N}_\mathcal{E}$ as:

$$\begin{aligned} \mathcal{N}_\mathcal{E} &: [0, \mathcal{E}_{\max}] \to [-1,\ 1],\ x \mapsto \frac{1}{\alpha_\mathcal{E}} \log_e(x+1) - 1, \\ \mathcal{N}_\mathcal{E}^{-1} &: [-1,\ 1] \to [0, \mathcal{E}_{\max}],\ \bar{x} \mapsto \exp(\alpha_\mathcal{E}(\bar{x}+1)) - 1, \end{aligned} \tag{4.59}$$

where $\mathcal{E}_{\max}$ denotes to maximal strain energy (within the space of admissible displacements) and with $\alpha_\mathcal{E} = \log_e(1+\mathcal{E}_{\max})/2$. The shape parameter space $\mathcal{P}$ is also normalized, by using a standard affine min–max normalization onto the interval $[-1,\ 1]$, i.e., we define the normalization function $\mathcal{N}_\mu$ as:

$$\mathcal{N}_\mu : \mathcal{P} \to [-1,\ 1]^{N_p}, \boldsymbol{\mu} \mapsto \frac{1}{\alpha_\mu}(\boldsymbol{\mu} - \beta_\mu), \tag{4.60}$$

where $\alpha_\mu = (\max(\mathcal{P}) - \min(\mathcal{P})/2$ and $\beta_\mu = (\max(\mathcal{P}) + \min(\mathcal{P}))/2$.

Let us write the strain energy as approximated by the PANN with these additional normalization steps:

$$\mathcal{E}_\theta^{\mathrm{NN}}(\mathbf{u};\boldsymbol{\mu}) = (\mathcal{N}_\mathcal{E}^{-1} \circ \bar{\mathcal{D}}_\theta^{\mathrm{NN}})\Big((\mathcal{N}_\mathcal{E} \circ \boldsymbol{\mathcal{E}}_{\mathrm{ref}})(\mathbf{u}); \mathcal{N}_\mu(\boldsymbol{\mu})\Big). \tag{4.61}$$

One would require the partial derivatives of the strain energy $\mathcal{E}_\theta^{\mathrm{NN}}$ with respect to both variables $\mathbf{u}$ and $\boldsymbol{\mu}$. The first one is needed to evaluate the applied (or reaction) loads as needed when evaluating the static equilibrium of a lattice structure with the DDM, and the second one is needed for shape optimization later on. With the normalization step, the applied load (as initially given by Eq. (4.57)) reads as:

$$\mathbf{f}_\theta^{\mathrm{NN}}(\mathbf{u};\boldsymbol{\mu}) = \mathcal{E}_\theta^{\mathrm{NN}}(\mathbf{u};\boldsymbol{\mu}) \left[\bar{\mathbf{f}}_{\mathrm{ref}}^1(\mathbf{u}) \quad \cdots \quad \bar{\mathbf{f}}_{\mathrm{ref}}^{N_{\mathrm{ref}}}(\mathbf{u})\right] \frac{\partial \bar{\mathcal{D}}_\theta^{\mathrm{NN}}}{\partial \bar{\mathbf{e}}}\Big((\mathcal{N}_\mathcal{E} \circ \boldsymbol{\mathcal{E}}_{\mathrm{ref}})(\mathbf{u}); \mathcal{N}_\mu(\boldsymbol{\mu})\Big) \tag{4.62}$$

where:

$$\bar{\mathbf{f}}_{\mathrm{ref}}^i(\mathbf{u}) = \frac{1}{1+\mathcal{E}_{\mathrm{ref}}^i(\mathbf{u})}\mathbf{S}(\boldsymbol{\mu}_{\mathrm{ref}}^i)\mathbf{u}, \quad i = 1, \ldots, N_{\mathrm{ref}}.$$

The derivative w.r.t. the shape parameters $\boldsymbol{\mu}$ reads, with the normalizations, as:

$$\frac{\partial \mathcal{E}_\theta^{\mathrm{NN}}}{\partial \boldsymbol{\mu}}(\mathbf{u};\boldsymbol{\mu}) = \frac{\alpha_\mathcal{E}}{\alpha_\mu}\mathcal{E}_\theta^{\mathrm{NN}}(\mathbf{u};\boldsymbol{\mu})\frac{\partial \bar{\mathcal{D}}_\theta^{\mathrm{NN}}}{\partial \bar{\boldsymbol{\mu}}}\Big((\mathcal{N}_\mathcal{E} \circ \boldsymbol{\mathcal{E}}_{\mathrm{ref}})(\mathbf{u}); \mathcal{N}_\mu(\boldsymbol{\mu})\Big). \tag{4.63}$$

Again, the partial derivative $\partial \bar{\mathcal{D}}_\theta^{\mathrm{NN}}/\partial \bar{\mathbf{e}}$ and $\partial \bar{\mathcal{D}}_\theta^{\mathrm{NN}}/\partial \bar{\boldsymbol{\mu}}$ can be efficiently computed thanks to AD.

Remark. It is worth noting that in some PINN or PANN implementations, specific activation fonctions can be employed in the output layer to enforce physically motivated constraints on the network predictions. For instance, non–negative activation functions such as `ReLU` or `SoftPlus` are often used when the target quantity is constrained to be strictly positive, as in the case of strain energy or density fields. However, some studies observe that while they preserve solution positivity, they lead to significantly higher errors, up to two orders of magnitude worse than classic PINNs without such constraints (W. Zhang and Al Kobaisi 2022). In the present formulation, this explicit constraint is not required, since the logarithmic transformation of the energy inherently enforces positivity.

Physics-Consistent Loss Function

A second key element in embedding physical knowledge into the PANN architecture is the design of the loss function. The loss function plays a central role in the training

process, as it drives the optimization of the NN. In classical PINN formulations, the loss function combines data-driven terms with the residuals of the governing equations, ensuring that the model simultaneously learns from both physical laws and observed data. Our approach follows the same conceptual framework for constructing the loss, but since the governing equations are not explicitly known, the formulation relies entirely on data-driven terms derived from physically consistent simulation data.

The loss function of the proposed PANN is composed of four terms: one accounting for the energy prediction, two corresponding to the reaction force and moment vectors, and one related to the gradient of the strain energy w.r.t. the shape parameters, defined as:

$$\mathcal{L} = \mathcal{L}_{\mathcal{E}} + \mathcal{L}_{\mathbf{f}} + \mathcal{L}_{\mathbf{m}} + \mathcal{L}_{\partial\boldsymbol{\mu}} \tag{4.64}$$

where $\mathcal{L}_{\mathcal{E}}$, $\mathcal{L}_{\mathbf{f}}$, $\mathcal{L}_{\mathbf{m}}$, and $\mathcal{L}_{\partial\boldsymbol{\mu}}$ denote the contributions associated with the energy, reaction forces, reaction moments, and the gradient of the strain energy w.r.t. the shape parameters, respectively.

The loss function is evaluated over a set of precomputed data. These data are obtained from a set of N_{data} sample data points $(\mathbf{u}^i; \boldsymbol{\mu}^i) \in \mathbb{R}^{N_u} \times \mathcal{P}$:

$$\mathcal{I}_{\text{data}} = \left\{(\mathbf{u}^1; \boldsymbol{\mu}^1), \ldots, (\mathbf{u}^{N_{\text{data}}}; \boldsymbol{\mu}^{N_{\text{data}}})\right\} \tag{4.65}$$

For each data point, we precompute four quantities, namely the strain energy, the associated force and moment vectors, and the derivatives of the strain energy w.r.t. the shape parameters:

$$\mathcal{E}^i_{\text{data}} = \mathcal{E}(\mathbf{u}^i; \boldsymbol{\mu}^i), \quad \mathbf{f}^i_{\text{data}} = \mathbf{f}(\mathbf{u}^i; \boldsymbol{\mu}^i), \quad \mathbf{m}^i_{\text{data}} = \mathbf{m}(\mathbf{u}^i; \boldsymbol{\mu}^i), \quad \boldsymbol{\delta}_\mu \mathcal{E}^i_{\text{data}} = \Delta^h_\mu[\mathcal{E}](\mathbf{u}^i; \boldsymbol{\mu}^i).$$

It seems not straightforward how to compute the derivatives w.r.t. the shape parameters analytically. Therefore, we use central finite difference to evaluate these gradients approximately (this computation step is denoted $\Delta^h_\mu[\mathcal{E}]$). Finally, this leads to a data set of the form:

$$\mathcal{S}_{\text{data}} = \left\{(\mathcal{E}^1_{\text{data}}, \mathbf{f}^1_{\text{data}}, \mathbf{m}^1_{\text{data}}, \boldsymbol{\delta}_\mu \mathcal{E}^1_{\text{data}}), \ldots, (\mathcal{E}^{N_{\text{data}}}_{\text{data}}, \mathbf{f}^{N_{\text{data}}}_{\text{data}}, \mathbf{m}^{N_{\text{data}}}_{\text{data}}, \boldsymbol{\delta}_\mu \mathcal{E}^{N_{\text{data}}}_{\text{data}})\right\}. \tag{4.66}$$

With these notations in hand, let us detail the different terms involve in the loss function. First, the energy loss is defined as:

$$\mathcal{L}_{\mathcal{E}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(\bar{\mathcal{E}}^{\text{NN}}_\theta(\mathbf{u}^i; \boldsymbol{\mu}^i) - \bar{\mathcal{E}}^i_{\text{data}}\right)^2. \tag{4.67}$$

Since the reaction forces and moments span multiple orders of magnitude, a relative error is used to ensure that the model does not disproportionately weight larger values. The reaction force and moment losses are defined as:

$$\mathcal{L}_{\mathbf{f}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \frac{\left\| \mathbf{f}_{\theta}^{\text{NN}}(\mathbf{u}^i; \boldsymbol{\mu}^i) - \mathbf{f}_{\text{data}}^i \right\|_2^2}{\|\mathbf{f}_{\text{data}}^i\|_2^2}, \tag{4.68}$$

$$\mathcal{L}_{\mathbf{m}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \frac{\left\| \mathbf{m}_{\theta}^{\text{NN}}(\mathbf{u}^i; \boldsymbol{\mu}^i) - \mathbf{m}_{\text{data}}^i \right\|_2^2}{\|\mathbf{m}_{\text{data}}^i\|_2^2}, \tag{4.69}$$

where $\mathbf{f}_{\theta}^{\text{NN}}$ and $\mathbf{m}_{\theta}^{\text{NN}}$ are the predicted applied forces and moments which are the two sub-vectors of $\mathfrak{f}_{\theta}^{\text{NN}}$. For the gradient w.r.t. the shape parameters, the corresponding loss term is defined as:

$$\mathcal{L}_{\partial\boldsymbol{\mu}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left\| \frac{\partial \mathcal{E}_{\theta}^{\text{NN}}}{\partial \boldsymbol{\mu}}(\mathbf{u}^i; \boldsymbol{\mu}^i) - \boldsymbol{\delta}_{\mu} \mathcal{E}_{\text{data}}^i) \right\|_2^2. \tag{4.70}$$

In conclusion, the loss function is a diverse combination of terms that account for multiple aspects of the problem, including energy consistency, reaction forces and moments, and the sensitivity of the system to geometrical parameters.

4.4.3 Efficient Dataset Generation

A crucial and time-consuming step is the generation of the dataset. The dataset must be sufficiently large and diverse to enable the NN to learn the underlying mechanical behavior of parametrized cells. Sampling strategies are diverse, and their choice significantly influences the dataset's quality, computational cost, and representativity. In this work, two different strategies were investigated, as illustrated in Figure 4.12.

The first strategy, referred to as the in-situ sampling method, aims to mimic the actual usage of the NN by generating data under realistic operating conditions. This method consists in constructing a $3 \times 3 \times 3$ block of cells, or potentially a more complex arrangement, where each cell is assigned independent geometric parameters sampled from a uniform distribution. Random boundary conditions are then applied on the structure, and the equilibrium problem is solved using the DDM. During the resolution process, all pairs of imposed displacements and corresponding reaction forces and moments at the cell boundaries are recorded to build the dataset. The key strength of this approach is its strong physical consistency, as it accurately reproduces the conditions experienced by the surrogate model during the online phase.

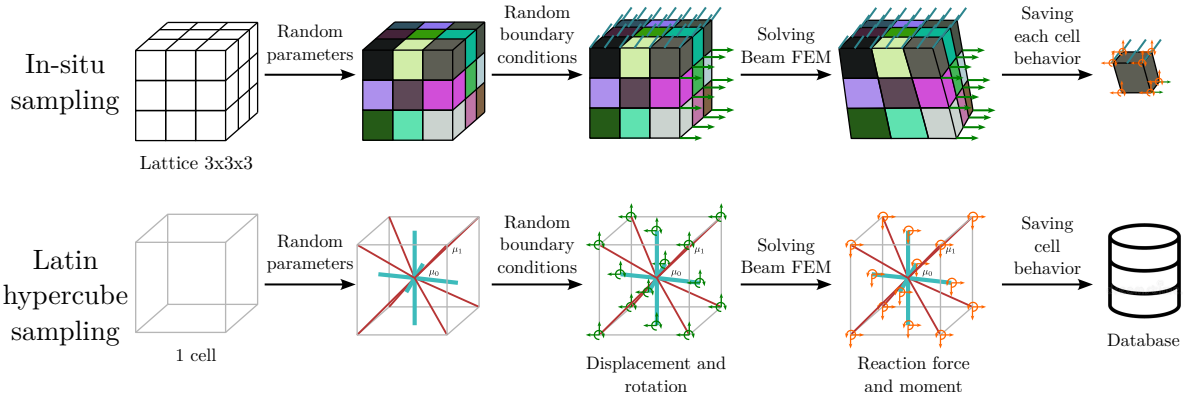

Fig. 4.12 Overview of the sampling strategies explored for dataset generation. The first strategy relies on LHS to sample the design space, while the second one consists in extracting cell loading cases from full-scale simulations based on global boundary conditions.

The second approach relies on a LHS of the design space. This statistical sampling technique ensures a uniform coverage of the parameter domain by dividing it into equally probable intervals and selecting one random sample within each. Geometric parameters and boundary displacements are sampled independently using this strategy, allowing an efficient and simple generation of large datasets. Compared to random sampling, LHS reduces the risk of clustering—regions where samples tend to concentrate—thereby improving the uniformity of the parameter space coverage. However, since no structural analysis is involved during the sampling, the generated data points may not always be representative of the physical conditions the model will experience during deployment. In summary, the in-situ sampling method provides higher physical fidelity by incorporating actual loading conditions, while LHS offers a faster and more uniform exploration of the design space.

Three main issues arise with the in-situ sampling method, ultimately preventing its use in the present study. The first issue concerns the computation of the gradient of the energy with respect to the geometric parameters, which is required for the loss term $\mathcal{L}_{\partial\boldsymbol{\mu}}$. Incorporating this computation into the in-situ sampling strategy is not straightforward, as it requires perturbing the geometrical parameters and recomputing the reaction forces and moments for each perturbation, which substantially increases the algorithmic complexity. The second issue relates to the complex dataset post-processing. Due to the iterative nature of the DDM, the displacement updates progressively decrease in magnitude throughout the resolution process, as the interface equilibrium is gradually achieved and convergence is approached. Consequently, a significant part of the dataset is dominated by near-zero displacements, resulting in correspondingly small energy values.

Training a NN over a dataset spanning several orders of magnitude is known to hinder convergence and degrade predictive performance. Finally, the method suffers from a lack of computational efficiency. A given cell configuration may appear multiple times in the dataset, under the same loading conditions. This redundancy leads to unnecessary recomputations of the same mechanical response at the microscale. Moreover, the construction of the 3D structure and the generation of diverse boundary conditions for each global simulation adds overhead that is not required in the LHS-based sampling strategy. All datasets used in this work are generated using the second sampling strategy, which is based on LHS, with parameter ranges summarized in Table 4.6.

Table 4.6 Sampling parameters used in the LHS-based dataset generation.

[1] The Young's modulus was determined experimentally, as described in Section 3.1.3.
[2] The bounds are chosen to satisfy the small displacement and rotation assumption associated with the linear beam model.
[3] All quantities are expressed in arbitrary but internally consistent units ("a.u.").

Parameter	**Value**
Cell size	1 a.u.[3]
Young's modulus E	1013 MPa[1]
Poisson's ratio ν	0.3
Geometric parameters $\boldsymbol{\mu}$	$[0.01,\ 0.1]$ a.u.
Translational displacement components u_x, u_y, u_z[2]	$[-0.1,\ 0.1]$ a.u.
Rotational displacement components $\theta_x, \theta_y, \theta_z$[2]	$[-0.01,\ 0.01]$ a.u.

Accelerated Dataset Generation Method.

This study focuses exclusively on linear elastic mechanical behavior, a property that can be leveraged to accelerate the sampling process. Thanks to the linearity of the constitutive model, the dataset can be augmented by scaling both the displacement vector and the corresponding reaction force and moment vector by a scalar factor α. This procedure enables the generation of additional physically consistent data points without requiring any new simulations, thereby offering a significant computational advantage. Moreover, it produces linearly dependent data points, which improves the conditioning of the dataset and facilitates the convergence of the NN. While the scaling factor α could be sampled from a uniform distribution, it is more advantageous to sample it from a logarithmic distribution. This ensures a wider and more balanced coverage of the energy scale, which is particularly relevant for improving the network's robustness across several orders of magnitude.

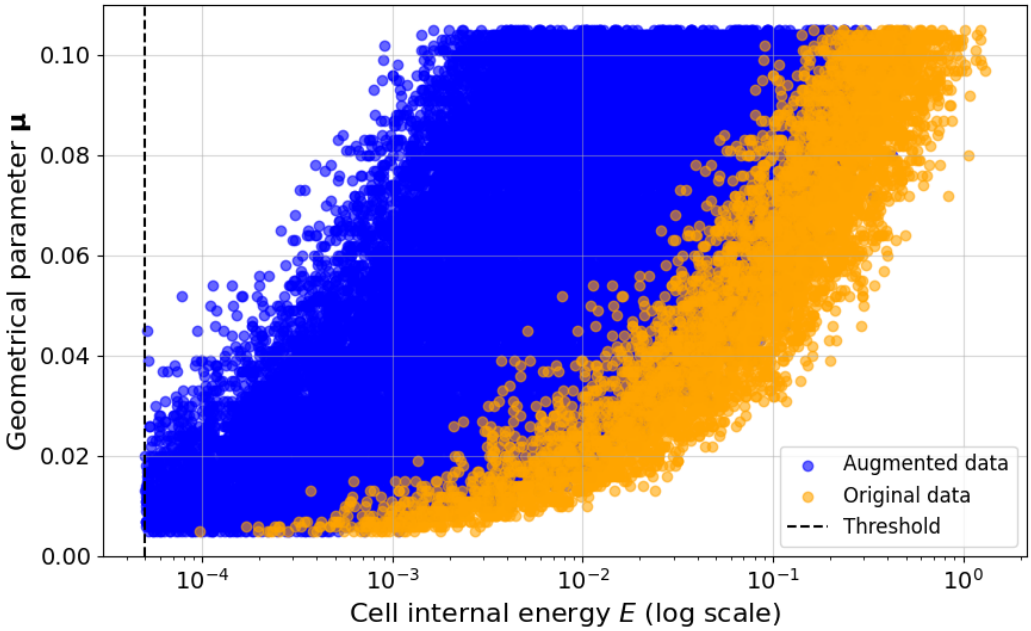

Fig. 4.13 Example of the distribution of original and augmented data in the $(\mathcal{E}, \boldsymbol{\mu})$ space with energy threshold filtering for a single geometrical parameter.

Remark. Due to the quadratic dependence of the internal energy on the displacement vector in linear elasticity, the gradient of the energy with respect to the radius must be scaled by α^2.

A second post-processing step is required to limit the presence of very small energy values in the dataset. To this end, a threshold is applied to the computed energy, and all data points with an energy value below this threshold are discarded. The augmentation process parameters chosen are summarized in Table 4.7.

Table 4.7 Parameters of the augmentation process.

Parameter	**Value**
Scaling factor range	$[10^{-1}, 10^{0}]$
Scaling factor distribution	Logarithmic
Number of augmentations per sample	5
Scaling values	$[0.1,\ 0.158,\ 0.251,\ 0.398,\ 0.631]$
Energy threshold	$5 \cdot 10^{-5}$

An example of the dataset distribution in the $(\mathcal{E}, \boldsymbol{\mu})$ space is shown in Figure 4.13. The original dataset corresponding to a single geometrical parameter comprises 5,000

samples generated using LHS. This dataset is subsequently augmented to 25,000 samples and further refined by applying an energy threshold, resulting in a final dataset of 29,230 samples. One and a half hours of computational time on a single CPU core were required to generate this dataset. However, it is important to note that the dataset generation process is parallelizable, and the computational time can be significantly reduced by leveraging multiple CPU cores. The generation of the dataset has been implemented in Python using the `FEniCSx` library (Alnæs et al. 2014; Baratta et al. 2023; Scroggs, Baratta, et al. 2022; Scroggs, Dokken, et al. 2022).

Remark. A specific point worth emphasizing concerns the computation of the gradient of the internal energy with respect to the geometric parameters. In principle, this quantity can be evaluated analytically directly within the FE solver. However, the use of the beam penalization method introduced in Chapter 3 introduces a non-trivial dependency of the stiffness on the geometry, which complicates the derivation of analytical gradients. As a result, a centered finite difference approximation is employed to estimate the derivative of the internal energy with respect to the geometrical parameters.

4.4.4 Training Procedure and Evaluation of the PANN

NN training is a multifactor process that depends on numerous hyperparameters and architectural decisions. All developments rely on a PANN implemented in `PyTorch` (Paszke et al. 2019). The network is optimized with AdamW optimizer, which augments Adam optimize with a decoupled weight decay term, thereby improving regularization and generalization (Loshchilov and Hutter 2019). In order to investigate the behavior of the PANN surrogate model within the framework of the domain decomposition method, a network was trained using a geometry with one single shape parameter. The training relied on the dataset presented in Figure 4.13 and the hyperparameters listed in Table 4.8. The computations were carried out on a laptop equipped with an Intel i7-12850HX CPU and an NVIDIA RTX A3000 Laptop GPU with 12,GB of memory.

Training campaigns were repeated under different configurations to seek a practical balance between accuracy, consistency, and computational resources. Table 4.9 summarizes the numerical performance of the PANN for different network architectures and dataset sizes.

Each configuration required approximately ten hours of training. The best-performing model, composed of five hidden layers with 64 neurons per layer, was selected for further analysis. In this study, only the dataset size, network depth, and width were varied, while all other hyperparameters were kept constant. A broader exploration of the hyperparameter space would be necessary to identify the truly optimal configuration. The

Table 4.8 Parameters for the training process.

Parameter	**Value**
Activation function	`tanh`
Batch size	256
Number of epochs	10000
Learning rate	3×10^{-3}
Learning decay	0.8
Optimizer	AdamW (Loshchilov and Hutter 2019)
Initialization method	Xavier normal (Glorot and Bengio 2010)
No. Geometric param.	1
Greedy precision $\varepsilon_{\text{greedy}}$	10^{-6}
No. Inputs	6 $(1 : \boldsymbol{\mu} + 5 : \mathcal{E}^i_{\text{ref}})$
%Validation set	10%

Table 4.9 Numerical performance of the PANN for different network architectures and dataset sizes. The errors ε_E, ε_f, ε_m, and $\varepsilon_{\partial\mu}$ correspond to the relative error of each loss term, as defined in Equation (4.64). The best result for each metric is highlighted in yellow.

No. Layers	**No. Neurons per Layer**	$\boldsymbol{\varepsilon_E}$	$\boldsymbol{\varepsilon_f}$	$\boldsymbol{\varepsilon_m}$	$\boldsymbol{\varepsilon_{\partial\mu}}$
1 geometries, $\varepsilon_{\text{greedy}} = 10^{-6}$, 5,938 evaluation points					
3	32	3.37×10^{-3}	3.36×10^{-3}	1.55×10^{-2}	2.63×10^{-3}
1 geometries, $\varepsilon_{\text{greedy}} = 10^{-6}$, 29,230 evaluation points					
3	64	8.29×10^{-4}	1.19×10^{-3}	6.98×10^{-3}	1.72×10^{-3}
3	128	8.71×10^{-4}	1.50×10^{-3}	7.00×10^{-3}	1.74×10^{-3}
4	64	8.46×10^{-4}	1.12×10^{-3}	6.53×10^{-3}	1.17×10^{-3}
5	64	3.05×10^{-4}	6.44×10^{-4}	6.47×10^{-3}	1.09×10^{-3}
5	128	4.02×10^{-4}	9.65×10^{-4}	7.00×10^{-3}	1.05×10^{-3}

evolution of the loss during training for the selected model is presented in Figure 4.14(a). The loss decreases steadily throughout the epochs before reaching a plateau towards the end of training, indicating that the model successfully extracted the underlying patterns from the dataset. After the completion of training, the model performance was assessed on both the training and validation datasets. Figure 4.14(b) reports the relative errors associated with each loss term for all data samples. Each point corresponds to a single data sample, with the color indicating the value of the geometric parameter $\boldsymbol{\mu}$. The results confirm that the model attains high accuracy across all metrics, with most errors

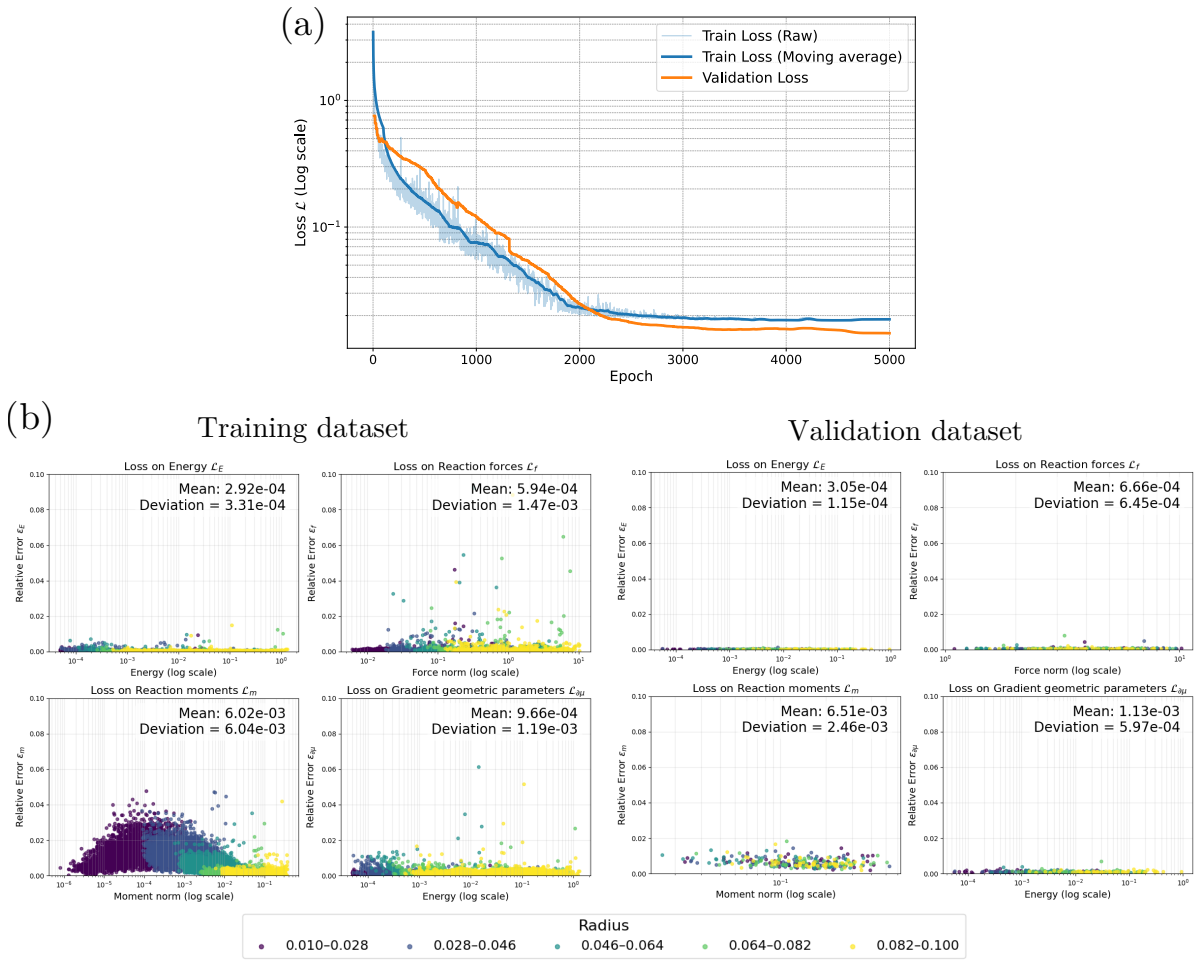

Fig. 4.14 Results of the trained PANN. (a) Evolution of the training and validation losses over the epochs. (b) Relative errors for each loss term evaluated over the entire training and validation datasets. Each marker corresponds to a single data sample, and the color encodes the value of the geometric parameter $\boldsymbol{\mu}$.

remaining below 1%. The moment predictions exhibit slightly larger deviations, reaching up to 5% relative errors for smaller values of $\boldsymbol{\mu}$. This behavior can be attributed to the higher range of absolute magnitude of the moments, which, after normalization, leads to a stronger compression of the data range. Consequently, the learning process becomes more challenging, resulting in reduced precision for this quantity.

Application of the PANN to Structural Benchmarks

After training, the PANN surrogate model is integrated into the DDM framework to solve full-scale structural problems. Each benchmark problem presented in Section 4.1.2 is addressed using the DDM framework enhanced with the PANN surrogate model.

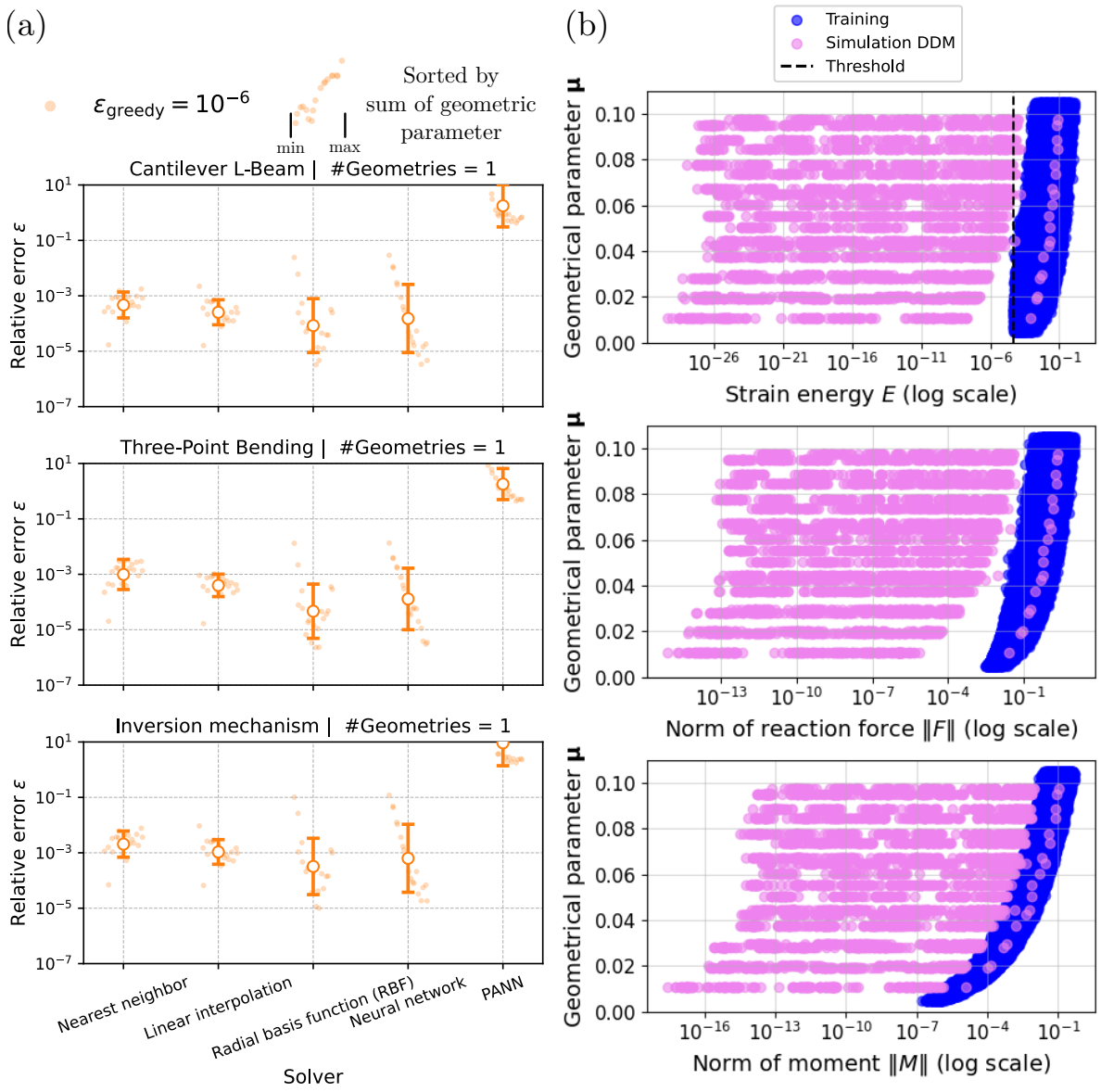

Fig. 4.15 PANN used on full-scale benchmark problems. (a) Relative error on the displacement field, and (b) evaluation points of the PANN during the solution of full-scale Cantilever L-beam benchmark problem.

Figure 4.15(a) reports the relative error on the displacement field for each benchmark problem. The errors obtained with the PANN architecture are completely out of scale compared to the previous methods. To understand the origin of this poor performance, it is necessary to analyze the evaluation points of the PANN during the solution of the full-scale problems. Figure 4.15(b) illustrates the evaluation points of the PANN during the resolution of the Cantilever L-beam problem. A clear trend observed in the results is that the training domain does not overlap with the evaluation points during the

solving phase. Since NN are generally unable to extrapolate beyond their training range, it is not surprising that the relative errors on the displacement field are significantly higher than with the other approaches. The points located within the training range typically correspond to cells close to the boundary conditions, where the energy is larger during the first iterations of the CG. The points observed at the lowest values result from the convergence process of the CG algorithm. As explained in Section 4.2.3, the CG algorithm evaluates the model along a search direction, which is a linear combination of the current residual and the previous search direction. As the iterations progress, this direction decreases in magnitude and tends towards zero at convergence. It is therefore important to note that the search direction does not represent a physical displacement field. Consequently, it is expected that the PANN is frequently evaluated outside its training range, which explains the poor generalization and the large errors observed in the full-scale problems. This highlights the fact that, although the NN is well trained and performs accurately, its effectiveness can be significantly degraded if the simulation method in which it is embedded is not consistent with the assumptions underlying the network.

4.4.5 Nonlinear CG Solver for PANN-Based DDM

Nonlinear Conjugate Gradient Algorithm

To overcome the limitations of the linear CG method when combined with our surrogate model, we propose the use of the Nonlinear Conjugate Gradient (NCG) method. The NCG method generalizes this framework to the minimization of a general nonlinear functional (Hager and H. Zhang 2006; Karimi and Vavasis 2024)

$$\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}, \tag{4.71}$$

Starting from an initial approximation $\mathbf{x}_0 \in \mathbb{R}^n$, the nonlinear conjugate gradient method iteratively updates the sequence $\mathbf{x}_k$, $k \geq 1$, following the recurrence relation:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{4.72}$$

where the positive step size α_k is obtained by a line search, and the directions $\mathbf{d}_k$ are generated by the rule:

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \qquad \mathbf{d}_0 = -\mathbf{g}_0. \tag{4.73}$$

with β_k ensuring a form of conjugacy between successive directions. A widely used choice is the Polak–Ribière update (Polak and Ribiere 1969)

$$\beta_k^{\mathrm{PRP}} = \frac{\mathbf{g}_k^\top (\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}}. \tag{4.74}$$

Since the problem is no longer quadratic, exact conjugacy is lost, and convergence relies on combining these updates with a line search that satisfies conditions such as Wolfe or Armijo, more information in Hager and H. Zhang (2006).

A key advantage in our context is that the NCG method requires evaluations only at physically meaningful points, namely the gradients $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, which in our case correspond to displacement data at the interface. Consequently, the nonlinear variant is more naturally aligned with physics-informed surrogates, as all evaluations remain within the space of physically interpretable fields.

Numerical Implementation and Results

The NCG algorithm was implemented within the DDM framework as a replacement for the classical CG solver. An Armijo backtracking line search (Chan–Renous-Legoubin and Royer 2022) was employed to guarantee a sufficient decrease of the functional at each iteration. Figure 4.16(a) shows the evaluation points of the PANN during the solution of the cantilever L-beam benchmark problems using the NCG method. Similar trends to those observed previously in Figure 4.15(b) can be identified: some points lie within the training range, while many others fall outside of it. Nevertheless, the distribution of evaluation points now spans fewer orders of magnitude than with the linear CG method, which indicates a better compatibility between the surrogate model and the optimization algorithm. Figure 4.16(b) reports the relative error on the displacement field for each benchmark problem. The errors obtained with the NCG method are reduced by approximately one order of magnitude compared to the linear CG. This improvement is attributed to the more physically consistent evaluation points used during the optimization process. Nevertheless, the achieved accuracy remains insufficient to match the performance of the other surrogate models presented earlier. The need for a PANN surrogate capable of handling wide variations across several orders of magnitude therefore persists, although the range to be addressed becomes more manageable. Another method was also investigated, in which each problem was solved using a shifted PANN. In other words, since the problem lies within the linear elasticity framework, each input displacement during the resolution was shifted into the training domain of the surrogate, and after calculation, the result was shifted back. This procedure ensures that the PANN

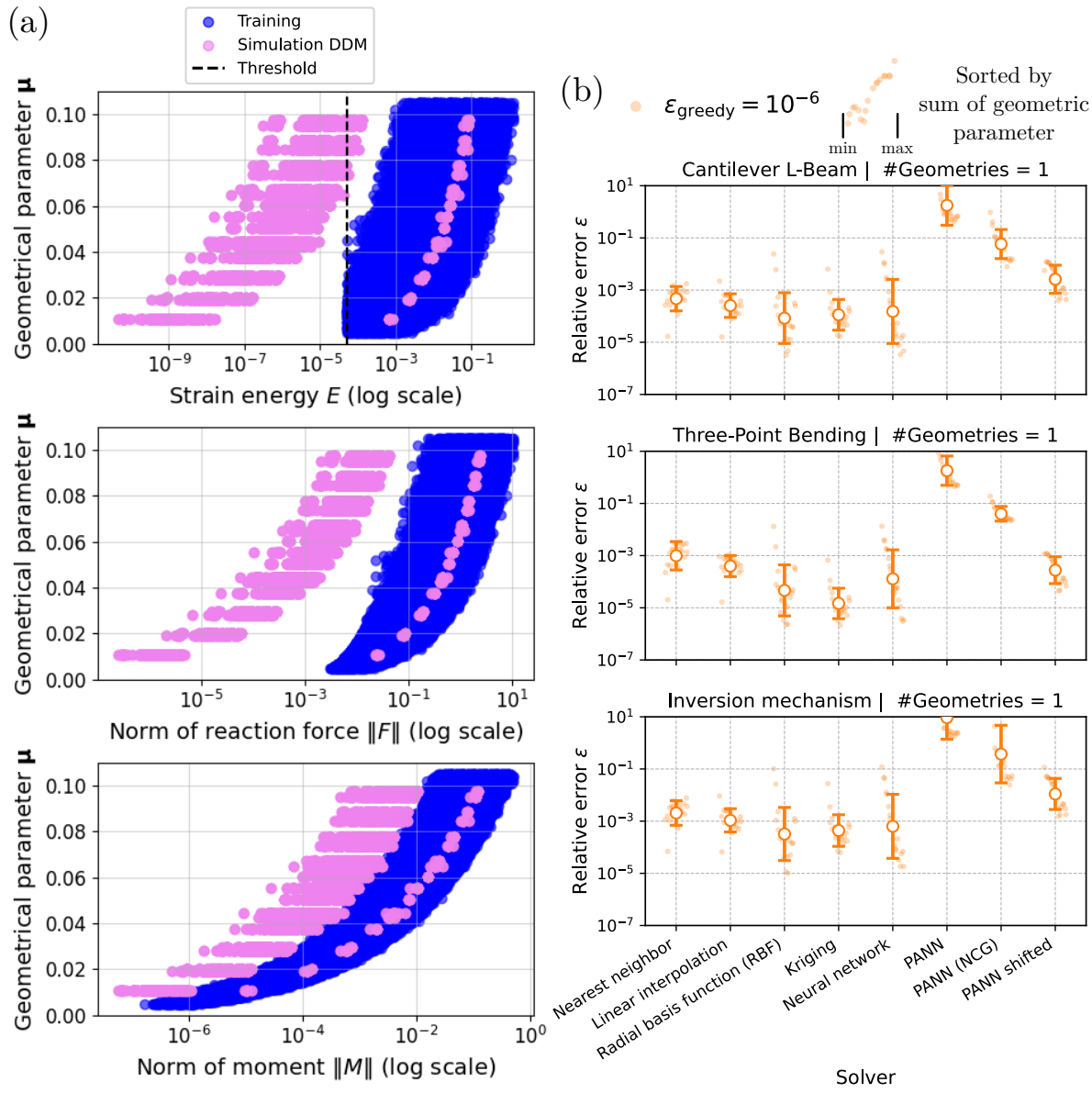

Fig. 4.16 Results of the NCG method in the context of PANN. (a) Evaluation points of the PANN with NCG algorithm during the solution of full-scale Cantilever L-beam benchmark problems, and (b) Relative error on the displacement field.

always operates within its training domain. The results obtained with this strategy are improved by an additional order of magnitude compared to the previous approach, with errors now below 1% for all benchmarks. In particular, for the three-point bending problem, the error decreases to about 0.1%. However, this method is not directly applicable in a nonlinear framework, as it would require re-linearizing the problem at every surrogate evaluation.

A natural idea to improve the performance of the PANN is to expand the training range so that it covers all possible displacement values that may occur during the CG iterations. This can be achieved either by further extending the dataset using the procedure described in Section 4.4.3, or by relying on the in situ sampling method. The drawback of such an approach is that the number of orders of magnitude that the PANN must learn to represent increases substantially. As a result, the model accuracy is expected to deteriorate as the range of the training data becomes larger, as discussed in Section 2.4.3. Several strategies can be considered to mitigate this limitation:

- Training multiple NN specialized for different ranges of orders of magnitude and selecting the appropriate model during the CG iterations (Deep Mixture of Experts (Eigen et al. 2014)). The main challenge with this approach lies in defining the criteria for switching between models and ensuring smooth transitions to avoid discontinuities in the predictions.
- Employing more advanced architectures, such as Neural ODEs (Kircher et al. 2023; Zanardi et al. 2023), in which the hidden states evolve according to an ordinary differential equation parameterized by a NN. This continuous-time formulation enables the network to better capture smooth dynamics and to adapt its effective depth to the complexity of the input.
- Reformulating the training data in terms of invariants (e.g., normalized energies or scale-independent quantities), so that the network only learns the reduced representation while the global scaling is reconstructed analytically (Klein et al. 2022). This approach can significantly compress the range of values to be learned, but it is only applicable in linear settings where such invariants can be clearly identified. For nonlinear behaviors, the lack of simple invariants makes this strategy impractical.

All these strategies, however, come at the price of increased algorithmic complexity and computational cost, and their applicability strongly depends on the underlying physical regime.

4.5 Discussion and Conclusion

Simulating lattice structures is computationally demanding, particularly when the cell geometry varies locally. This difficulty can be alleviated using the method introduced in Chapter 3, which relies on beam elements combined with penalization at junctions to

account for the lack of rigidity. Nevertheless, in the context of design optimization of multifunctional structures, the number of parameters grows significantly, resulting in an exponential increase in the number of required simulations. To address this challenge, the chapter introduces a novel approach that combines the DDM with a surrogate model, enabling efficient problem resolution while further providing the capability to compute gradients with respect to the geometrical parameters. The DDM enables the decomposition of the global problem into smaller subproblems, each corresponding to a unit cell in the case of lattice structures. This decomposition not only reduces the computational burden but also facilitates parallel processing, since each subproblem can be solved independently. The framework is further enhanced by introducing a surrogate model that approximates the mechanical response of each cell as a function of its geometrical parameters. The surrogate model relies on Schur complements, which capture the relationship between boundary displacements and the associated reaction forces and moments. Since evaluating the mechanical response of each cell is computationally expensive, a ROM is used to generate a reduced basis representation. A surrogate model is then derived from the reduced basis, enabling efficient interpolation throughout the design space. This surrogate model is reused across the DDM iterations, significantly accelerating the convergence process Two main families of surrogate models were explored: one based on learning the reduced coefficients of the Schur complement, and another directly learning the energy through a physics-augmented neural network approach.

For the first family, referred to as projection-based Reduced Order Model (pROM), several methods were investigated, ranging from the simplest nearest neighbor approach to more advanced interpolation techniques such as RBF and NN. The general strategy consists in first computing a reduced basis of Schur complements using a greedy algorithm, and then learning the mapping from the geometrical parameters to the reduced coefficients. The different surrogate models vary in terms of ease of implementation, computational cost, and predictive accuracy. It was observed that simple methods, such as nearest neighbor or linear interpolation, can already provide reasonable accuracy since the reduced coefficients typically evolve smoothly with the geometrical parameters. NN tend to outperform RBF-based approaches only when the dependence on the parameters becomes highly nonlinear. Overall, RBF interpolation exhibited the most consistent mean accuracy across all cases. The main limitation of this family of methods arises when dealing with nonlinear problems, since the Schur complement is no longer constant but depends on the displacement field $\mathbf{u}$, thereby requiring a more sophisticated surrogate model.

Table 4.10 Comparison of the different surrogate models presented in this chapter for the three-point bending benchmark case. The computation time is evaluated using the same random configuration as in Section 4.2, with 3 geometries and a greedy tolerance of $\varepsilon_{\text{greedy}} = 10^{-6}$. The preconditioner is constructed using a single Schur complement computed with the mean radius.

Method	Computation time (s)	Mean precision	Standard deviation	Class continuity
Reference methods				
Full Scale FEM	512.3	Reference	—	—
DDM standard	47.3	$\approx 10^{-6}$	—	—
Surrogate models based on α (pROM)				
Nearest Neighbor	29.9	9.86×10^{-4}	1.25	C^{-1}
Linear interpolation	19.2	3.97×10^{-4}	0.93	C^{0}
RBF	16.1	0.46×10^{-4}	2.25	C^{1}
NN	16.9	1.29×10^{-4}	2.56	C^{∞}
Surrogate model based on internal energy (PANN)				
PANN	18.3	1.8	1.29	C^{∞}
PANN (NCG)	17.4	4.0×10^{-2}	0.64	C^{∞}
PANN shifted	18.6	2.77×10^{-4}	1.18	C^{∞}

The second family of surrogate models, referred to as PANN, directly learns the internal energy of the cell as a function of both the geometrical parameters and the imposed boundary displacements. This approach is more general and can be extended to nonlinear problems, as it does not rely on the linearity of the Schur complement. The PANN is trained using a physics-augmented loss function that incorporates not only the error on the energy but also on the reaction forces and moments, which are derived from the energy using AD. This training strategy ensures that the surrogate model captures both the energetic and mechanical aspects of the cell's behavior. The PANN approach demonstrated high accuracy on the training and validation datasets, achieving relative errors below 1% for most quantities. However, when integrated into the DDM framework to solve full-scale problems, the performance exhibits a completely different behavior. The CG convergence method is inadequate in this framework, as its evaluation points have no physical meaning. To address this, the NCG approach is adopted, defining

search directions through displacement fields that retain a clear physical interpretation. This change led to a significant improvement in accuracy, reducing errors by an order of magnitude compared to the linear CG method. Despite these advancements, the PANN approach still faces challenges in achieving the same level of accuracy as the pROM methods, particularly due to the wide range of displacement values encountered during the DDM iterations. The main limitation of the PANN approach lies in the difficulty of training a NN to accurately represent a function that spans several orders of magnitude. The main advantage of pROM is that the accuracy of the surrogate is independent of the order of magnitude of the cell's energy and reaction forces.

In conclusion, the methodology presented in this chapter provides a promising avenue for the efficient simulation of lattice structures with locally tailored geometries. By combining the DDM with surrogate models, the computational cost can be significantly reduced while maintaining high accuracy. The flexibility of the framework to accommodate diverse geometries and unit-cell topologies makes it particularly well suited for design optimization tasks involving complex or conformal lattices (See Figure 4.1). Nevertheless, further developments are required to extend the methods to nonlinear problems and to improve the PANN approach, enabling it to better handle the wide range of displacement values encountered in the DDM framework. Potential strategies to address this challenge have been outlined in Section 4.4.4. Finally, Table 4.10 provides a comprehensive overview of the different methods discussed throughout this chapter, highlighting their respective computational costs and accuracy levels.

The following chapter will focus on the integration of this method within an optimization framework, leveraging the efficient simulation capabilities and gradient computations offered by the surrogate models.

Chapter 5

Towards Efficient Optimization of Locally Tailored Lattice Structures

"Nature builds strength with emptiness." — Anonymous

Contents

5.1 Formulation of the Optimization Framework 154
5.1.1 Optimization Problem 154
5.1.2 Surrogate Model for Relative Density 157
5.2 Optimization Results and Design Space Analyses 158
5.2.1 Effect of the Relative Density Constraint on Structural Performance 158
5.2.2 Post-Processing 160
5.2.3 Three-point Bending Beam Study Case with Different Parameter Spaces 161
5.2.4 Case Study: Full Optimization Pipeline Applied to a Femoral Head Implant 165
5.3 Experimental Validation on Benchmark Structures 168
5.3.1 Numerical Optimization Performance 168
5.3.2 Fabrication, Post-Processing, and Experimental Preparation 170
5.3.3 Experimental Results and Discussion 173
5.4 Concluding Remarks on the Optimization of Locally Tailored Lattice Structures 175

Lattice structure optimization is particularly challenging due to the strong coupling between geometry and performance, which leads to high-dimensional design spaces that must be explored to fully exploit their potential. Several strategies have been proposed to reduce the dimensionality of the design space while preserving the ability to capture high-performance designs. As previously discussed, graded structures are a great example, where only a few parameters, such as the beam radius or the cell size, are used to control the variation of properties across the entire structure. Nevertheless, when addressing complex geometries subjected to multiple loading conditions, these methods remain insufficient to meet the performance requirements of the structure. Moreover, the choice of the grading direction and the definition of variation parameters are usually imposed by the designer, which may significantly limit the achievable performance of the optimized structure. Allowing each cell to vary independently provides a much richer design space, but it also results in a considerably more complex optimization problem. Indeed, the number of design variables scales with the number of cells, and each cell can be characterized by multiple geometric parameters, such as the beam radius, the cell size, or even the topology. This high dimensionality significantly increases the computational cost of the optimization process.

5.1 Formulation of the Optimization Framework

We propose a novel optimization strategy built upon the simulation framework introduced in Chapter 4, thereby enabling the use of gradient-based optimization algorithms within a cell-independent design space. The objective is to identify the optimal geometric parameters of each cell composing the structure. In this study, the beam radii associated with the three primitive unit cells—referred to as hybrid geometries—are selected as design variables, as detailed in Section 4.1.1. Nevertheless, the methodology can be readily extended to other parameters, including more advanced ones such as cell morphing, which would allow the lattice to adapt to complex real-world applications.

5.1.1 Optimization Problem

The optimization objective is to enhance the global stiffness of the lattice structure while maintaining a prescribed mass (or relative density). In other words, the goal is to obtain a mechanically efficient structure that maximizes stiffness for a given amount of material. This objective is commonly formulated as the minimization of the compliance under a

relative density constraint. Mathematically, the problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{p}} \quad & C(\mathbf{p}) = \mathbf{F}^T \mathbf{u}(\mathbf{p}) \\ \text{s.t.} \quad & g_\rho(\mathbf{p}) \equiv \frac{1}{N_{cell}} \sum_e \rho_e(\boldsymbol{\mu}) - \bar{\rho} \leq 0, \\ & \boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\max}, \qquad \forall\, e \in \{1, \ldots, N_{cell}\}. \end{aligned} \tag{5.1}$$

where $g_\rho(\mathbf{p})$ denotes the constraint function associated with the relative density of the structure, N_{cell} the number of cells, $\boldsymbol{\mu}$ the design parameter of each geometry type in the cell e, $\rho_e(\boldsymbol{\mu})$ its relative density, and $\bar{\rho}$ the target average relative density. To ensure the continuity of the optimization problem and to avoid degenerate designs, the design variable vector $\boldsymbol{\mu}$ is bounded componentwise by $\boldsymbol{\mu}_{\min} = [0.01, \ldots, 0.01]$ and $\boldsymbol{\mu}_{\max} = [0.1, \ldots, 0.1]$. Since the proposed simulation framework was specifically designed to enable efficient gradient computation, the optimization problem is solved using the `SLSQP` algorithm from the `SciPy` library (Virtanen et al. 2020), which is a gradient-based method suitable for constrained problems. The code is implemented in the open-source framework pyLattice (see Appendix A).

Gradients Under linear elasticity, the gradient of the compliance with respect to the design variables can be computed using Clapeyron's theorem $C(\mathbf{p}) = 2E(\mathbf{p})$, with $E(\mathbf{p})$ the strain energy. Hence

$$\frac{\partial C(\mathbf{p})}{\partial \boldsymbol{\mu}} = 2\, \frac{\partial E(\mathbf{p})}{\partial \boldsymbol{\mu}}, \tag{5.2}$$

and, equivalently, using the stiffness sensitivity,

$$\frac{\partial C(\mathbf{p})}{\partial \boldsymbol{\mu}} = -\,\mathbf{u}^T\, \frac{\partial \boldsymbol{K}(\mathbf{p})}{\partial \boldsymbol{\mu}}\, \mathbf{u}. \tag{5.3}$$

The gradient computation is different depending on the surrogate model used to predict the structural response.

Gradient of PANN. The PANN surrogate model –introduced in Section 4.4 –provides an efficient framework to predict both the structural response and its sensitivity to geometric parameters. The network architecture is specifically designed to facilitate the computation of gradients with respect to the input parameters, as all operations are differentiable by construction. Consequently, the gradient of the strain energy with respect to the geometric parameters can be directly evaluated through AD, while consistently

accounting for the applied normalization. However, the predictive performance of the current surrogate model remains suboptimal. Since the present study focuses on linear problems, the pROM method combined with an RBF-based surrogate can be employed to achieve improved overall prediction accuracy.

Gradient of RBF. In the pROM presented in Section 4.3.2, the gradient of the compliance is computed using the chain rule.

$$\frac{\partial C(\mathbf{p})}{\partial \boldsymbol{\mu}} = \sum_i -\mathbf{u}^T \mathbf{S}(\boldsymbol{\mu}^i_{\text{ref}}) \frac{\partial \tilde{\alpha}_i(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \mathbf{u}, \tag{5.4}$$

where α^i are the reduced coefficients functions and n is the number of modes retained in the reduced basis. In the specific case of radial basis functions, the gradient of the coefficients can be computed analytically. The RBF interpolant is defined as

$$f(\boldsymbol{\mu}) \;=\; \sum_j w_j\, \phi(r_j) \;+\; c_0 \;+\; \boldsymbol{c}^\top \boldsymbol{\mu}, \qquad r_j = \|\boldsymbol{\mu} - \boldsymbol{\mu}_j\|_2, \tag{5.5}$$

where $\{\boldsymbol{\mu}_j\}$ denote the interpolation centers, w_j are the associated radial weights, and $c_0, \boldsymbol{c}$ correspond to the coefficients of the polynomial function. In the case of the thin-plate spline kernel, the function is in the following form:

$$\phi(r) = r^2 \log r, \tag{5.6}$$

and the gradient of the interpolant takes the form (see Appendix B for the complete derivation):

$$\nabla f(\boldsymbol{\mu}) \;=\; \sum_j w_j \Big(2 \log r_j + 1\Big)(\boldsymbol{\mu} - \boldsymbol{\mu}_j) \;+\; \boldsymbol{c}. \tag{5.7}$$

with the convention that the contribution vanishes when $r_j = 0$. Equation (5.7) highlights that the gradient of the interpolant is itself a linear combination of shifted vectors $(\boldsymbol{\mu} - \boldsymbol{\mu}_j)$, weighted by coefficients that depend only on the distance r_j. This analytical expression enables the efficient computation of sensitivities in optimization contexts, as both the function value and its gradient can be evaluated exactly without resorting to numerical differentiation.

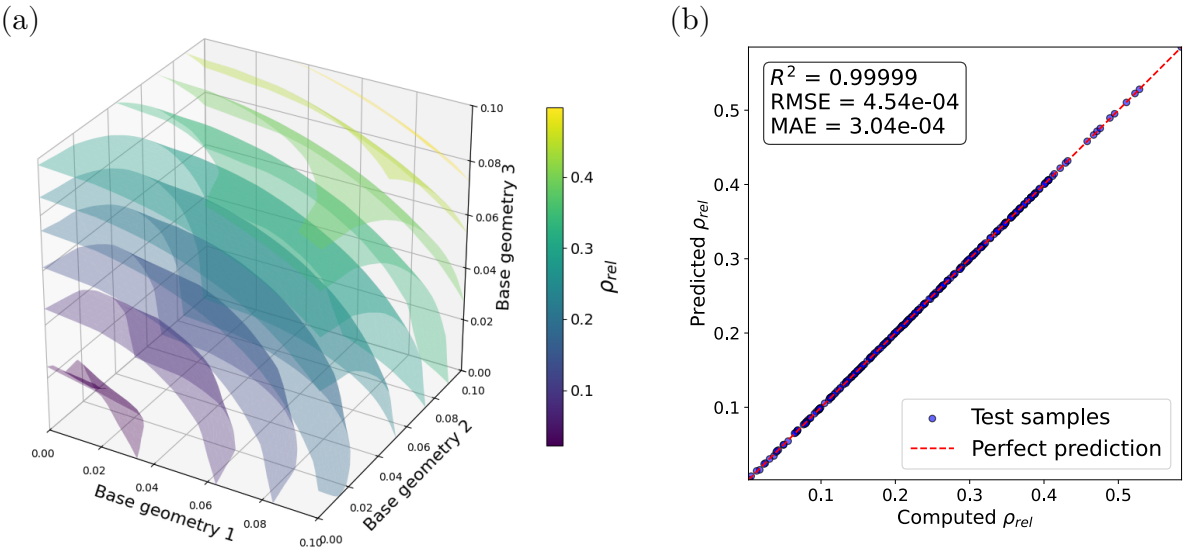

Fig. 5.1 Surrogate modeling of the relative density. (a) Iso-surfaces of relative density ρ_{rel} in the design space spanned by the three base geometries. The dataset is generated using full 3D meshing and numerical integration. (b) Parity plot comparing computed and Kriging-predicted relative densities on the test set. (R^2: coefficient of determination, RMSE: root-mean-square error, MAE: mean absolute error).

5.1.2 Surrogate Model for Relative Density

Constraining the relative density of lattice structures is essential to prevent trivial solutions in optimization problems. However, accurately computing the relative density of complex geometries can be computationally expensive, particularly when using a 1D beam model. In practice, an accurate evaluation typically requires generating a full 3D mesh of the geometry and performing numerical integration, which is time-consuming, especially in optimization contexts where the gradient of the relative density is also required. To address this issue, we propose employing a surrogate model to approximate the relative density of unit cells as a function of their geometric parameters (defined in Section 4.1.1). We assume a linear relationship between relative density and cell size, allowing the model to be applied to any specified cell size. A dataset is first generated by paving the design space with a step size of 0.01 for each parameter. The relative density of each configuration is then computed using a 3D mesh and numerical integration with the open-source library `Gmsh` (Geuzaine and Remacle 2009)[1]. Figure 5.1(a) illustrates the variation of the relative density ρ_{rel} with respect to geometric parameters. This visualization emphasizes the complex and non-linear relationship between the parameters

[1]Available in the open-source framework `pyLattice` see annexe A.

and the resulting relative density. This behavior results from the different numbers of beams within each geometry, as well as the interactions between them at the joints. Then, a surrogate model is trained to predict the relative density as a function of the geometric parameters. We adopt Kriging (Kleijnen 2009) as the surrogate model, owing to its ability to provide not only accurate predictions but also uncertainty estimates. Moreover, the Kriging model is well suited for interpolation tasks in low-dimensional spaces, making it an appropriate choice for this application, as only three parameters are considered in this example. Figure 5.1(b) presents the parity plot comparing the surrogate model predictions with the ground-truth values for a test set. The Kriging surrogate exhibits excellent predictive capability, as evidenced by a coefficient of determination close to unity ($R^2 \approx 1$) and very low prediction errors, with both the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) remaining below 5×10^{-4}. This surrogate model enables efficient evaluation of the relative density and its gradient during the optimization process, significantly reducing the computational cost compared to direct numerical integration.

5.2 Optimization Results and Design Space Analyses

5.2.1 Effect of the Relative Density Constraint on Structural Performance

Design constraints are a key component of any optimization framework, as they strongly influence the final outcome. Therefore, we investigate the effect of the relative density constraint on the optimized lattice structure. Figure 5.2(b) illustrates the optimized configurations for different target relative densities ($\bar{\rho} = 0.08$, 0.15, and 0.35) and for the unconstrained case at $\rho_{rel}^{dense} = 0.58$, under the cantilever L-beam benchmark described in Section 4.1.2. In each optimization, the most critical cells can be identified as those exhibiting the largest beam radii, which consistently correspond to the same locations within the structure across all relative density levels. Figure 5.2(a) reports on the left axis the normalized compliance as a function of the target relative density for both optimized and uniform designs. As expected, the compliance decreases with increasing relative density. However, across all density levels, the optimized designs consistently outperform their uniform counterparts. The relative improvement achieved through optimization is

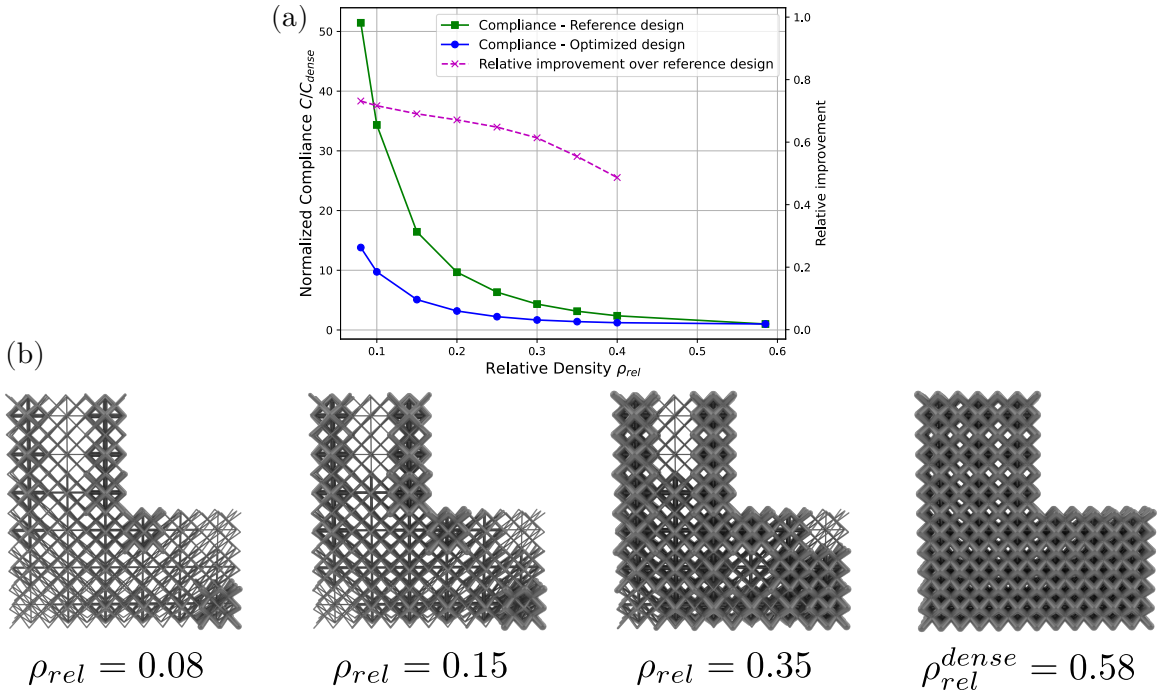

Fig. 5.2 Illustration of the dependencies of the relative density constraint on the optimized lattice structure under the cantilever L-beam study case, as presented in Section 4.1.2. (a) Normalized compliance as a function of the target relative density for optimized and uniform designs and relative improvement over reference design. (b) Optimized designs for different target relative densities ($\bar{\rho} = 0.08$, 0.15, 0.35, and without constraint at $\rho_{rel}^{dense} = 0.58$).

shown on the right axis of Figure 5.2(a) and is defined as:

$$\eta_{rel} = \frac{|C_{\text{opt}} - C_{\text{reference}}|}{|C_{\text{reference}}|}, \tag{5.8}$$

where C denotes the compliance. The results show that the optimization is particularly effective at low relative densities, where η_{rel} reaches high values. As the target relative density increases, the relative gain progressively decreases, indicating a reduced efficiency of the optimization at higher densities. Nevertheless, the compliance remains lower at small relative densities, reflecting the higher deformability of lighter structures. These results highlight that a balanced trade-off between structural performance and material savings can be achieved by selecting an intermediate target relative density.

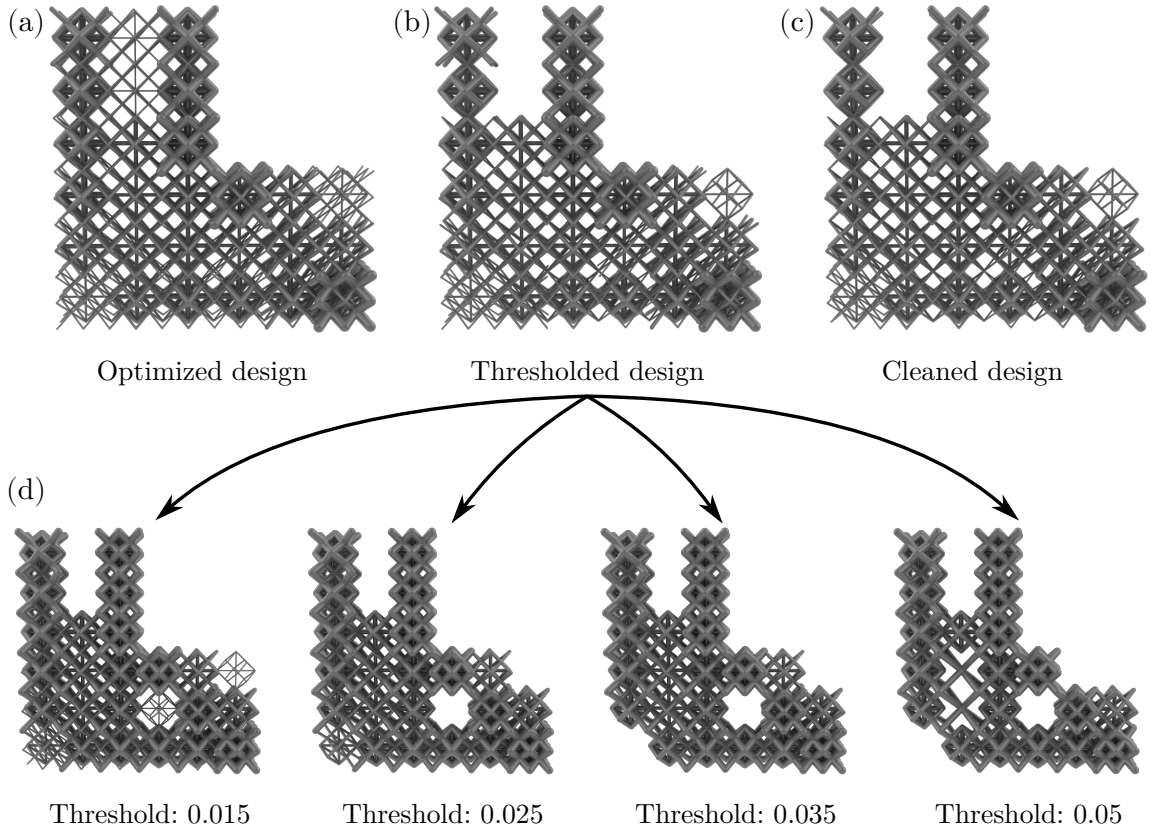

Fig. 5.3 Illustration of the post-processing procedure applied to the optimized lattice structure: (a) initial optimized design, (b) thresholded design after removal of beams whose radii are below the printing resolution capability, (c) cleaned design after elimination of non-load-carrying or isolated beams, and (d) designs obtained with different threshold values.

5.2.2 Post-Processing

After the optimization, a post-processing step is carried out to ensure both the printability and the structural integrity of the optimized lattice design. First, a threshold is applied to the beam radii in order to remove elements that are too thin to be printed reliably and that contribute negligibly to the overall mechanical response. This step is beneficial at two levels: first, it reduces the complexity associated with meshing the three-dimensional geometry, and second, it prevents fabrication-related issues, thereby ensuring that the optimized design remains printable. The choice of the threshold depends on both the 3D printing technique and the capability of the related machine, and should be calibrated individually for each case. Next, beams that do not participate in the structural stiffness are eliminated. These correspond to isolated elements that are not connected to other

beams or to any boundary and functional conditions, and therefore do not contribute to load transfer within the structure. By removing such elements, the final lattice becomes not only easier to print but also more mechanically consistent. Figure 5.3 illustrates the different stages of the post-processing procedure: (a) the initial optimized design, (b) the thresholded design after the elimination of beams whose radii are below the printing capability, and (c) the cleaned design after removal of non-load-carrying beams. Figure 5.3(d) shows the effect of varying the threshold value on the optimized design. The design changes significantly between different threshold values, highlighting the importance of this post-processing step in ensuring both the printability and the structural integrity of the final lattice structure. The choice of this parameter is primarily driven by manufacturing constraints and should therefore be selected accordingly.

5.2.3 Three-point Bending Beam Study Case with Different Parameter Spaces

Overview of Parameterization Methods

This section investigates several strategies for parameterizing the design space of the three-point bending beam benchmark introduced in Section 4.1.2. Wide possibilities exist to parameterize lattice structures, ranging from a single global parameter controlling the radius of all cells, to multiple independent parameters per cell, and including intermediate strategies such as functional gradients. In the following, representative methods are reviewed and compared to identify their respective advantages and limitations.

The relative density constraint is set to $\bar{\rho} = 0.15$ for all optimizations in order to enable a fair comparison of the results. Each optimization method is evaluated on different primitive unit cells, where each geometry introduced in Section 4.1.1 is tested individually, as well as in combination with the others, either two by two or all three together. The first and simplest method relies on a single parameter that uniformly controls the radius of all beams in the structure. Two cases are considered. In the first, all beams share a common radius, whereas in the second, each geometry type is assigned an independent radius, denoted by the subscript $_{\text{hyb}}$, thereby increasing the number of design variables. The second method applies a linear gradient to vary the radius along the beam length of the global structure, a well established technique in the literature (Y. Li et al. 2020), as discussed in Section 2.1.1. The gradient is characterized

by two parameters and defined as

$$\boldsymbol{\mu}(x) = \omega_1 x + \omega_0, \tag{5.9}$$

where x is the normalized coordinate along the structure length, and ω_0 and ω_1 denote the slope and constant term of the function, respectively. The direction of the gradient is aligned with the half-beam subjected to bending, which is generally considered the most effective strategy, as it directly targets the region most sensitive to deformation. Finally, the third method corresponds to the proposed approach, referred to as the cell-independent method, in which each cell, as well as each geometry within the cell, can vary independently.

Performance Analysis of Optimization Approaches

Table 5.1 summarizes the optimization setups considered in this study, together with their corresponding results. Moreover, Figure 5.4 shows the optimal designs obtained with each optimization method for the hybrid configuration composed of geometries [1, 2]. The findings indicate that the proposed approach, which allows each cell to vary independently, consistently outperforms the other methods in terms of compliance reduction. This advantage is particularly pronounced when multiple geometries are employed, as the optimization can leverage the strengths of each geometry to better accommodate the diverse loading conditions within the structure. However, this increased flexibility comes at the cost of a significantly larger number of design variables, leading to longer optimization times and a greater number of iterations required for convergence. Without the proposed method and the availability of gradients provided by the surrogate model, the computational cost would be prohibitive for such a high number of design parameters.

A noteworthy trend concerns the use of hybrid geometries. When comparing the results obtained with a single geometry to those achieved with two or three geometries, the constant case shows that hybridization yields performances that lie between those of the individual geometries. For instance, the combination of geometries 1 and 2 results in a compliance value between those obtained when each geometry is used alone. This suggests that the lower-performing geometry contributes equally to the design as the better-performing one, thereby limiting the overall performance of the hybrid configuration. Relaxing this constraint in the cases labeled with the suffix $_{\text{hyb}}$ highlights the benefits of allowing each geometry type to vary independently. This approach consistently improves performance compared to the use of a single parameter shared

Table 5.1 Summary of the optimization setups considered in this study. The second column specifies the cell geometries, where [1], [2], and [3] denote the geometries introduced in Section 4.1.1. Combinations such as [1, 2] correspond to the hybrid use of geometries 1 and 2, and similarly for the other cases. The suffix $_{\text{hyb}}$ denotes that the geometries within a cell are allowed to vary independently, whereas in the other cases all geometries share a common radius. The optimal parameters are defined as the beam radii associated with each geometry type for the constant configuration, and as the weighting coefficients of Eq. (5.9) for the linear gradient configuration. The last column indicates the optimization difficulty level, where more stars denote higher complexity. One star is assigned to the hybridization of geometries, one star to graded structures, and two stars to unit cell optimization.

[1] **Warning:** The maximum relative density of the cell has been reached, which means that the final density of the structure cannot achieve the target value $\bar{\rho}$.

Category	Geometries	No. Params	Compliance ($\times 10^{-3}$)	Opt. time (s)	Iterations	Opt. Params ($\times 10^{-2}$)	Complexity
Constant	[1][1]	—	6.854	—	—	[10.0]	
	[2]	1	5.019	—	—	[9.6]	
	[3]	1	2.363	—	—	[8.9]	
	[1, 2]	1	5.371	—	—	[7.4]	★
	[1, 3]	1	2.985	—	—	[6.8]	★
	[2, 3]	1	2.683	—	—	[4.4]	★
	[1, 2, 3]	1	4.229	—	—	[6.3]	★
	$[1, 2]_{\text{hyb}}$	2	4.486	217.2	25	[5.6, 9.1]	★
	$[1, 3]_{\text{hyb}}$	2	3.158	107.7	26	[9.1, 5.6]	★
	$[2, 3]_{\text{hyb}}$	2	2.252	64.9	18	[4.3, 7.8]	★
	$[1, 2, 3]_{\text{hyb}}$	3	3.363	84.8	14	[5.1, 2.0, 5.3]	★
Linear gradient along the beam length	[1][1]	—	6.854	—	—	—	
	[2]	2	4.743	3.1	6	[10.0, 7.3]	★
	[3]	2	2.225	7.3	8	[2.1, 8.3]	★
	[1, 2]	2	4.265	103.2	10	[5.7, 4.8]	★★
	[1, 3]	2	2.415	44.2	10	[9.1, 2.6]	★★
	[2, 3]	2	2.274	78.7	10	[4.3, 4.5]	★★
	[1, 2, 3]	2	3.096	720.9	15	[4.7, 2.4]	★★
Cell-independent	[1][1]	—	6.854	—	—	—	
	[2]	250	4.560	46.5	59	—	★★
	[3]	250	1.997	37.6	54	—	★★
	$[1, 2]_{\text{hyb}}$	500	2.618	1581	171	—	★★★
	$[1, 3]_{\text{hyb}}$	500	1.445	436.5	128	—	★★★
	$[2, 3]_{\text{hyb}}$	500	1.288	436.3	94	—	★★★
	$[1, 2, 3]_{\text{hyb}}$	750	1.043	19,171	240	—	★★★

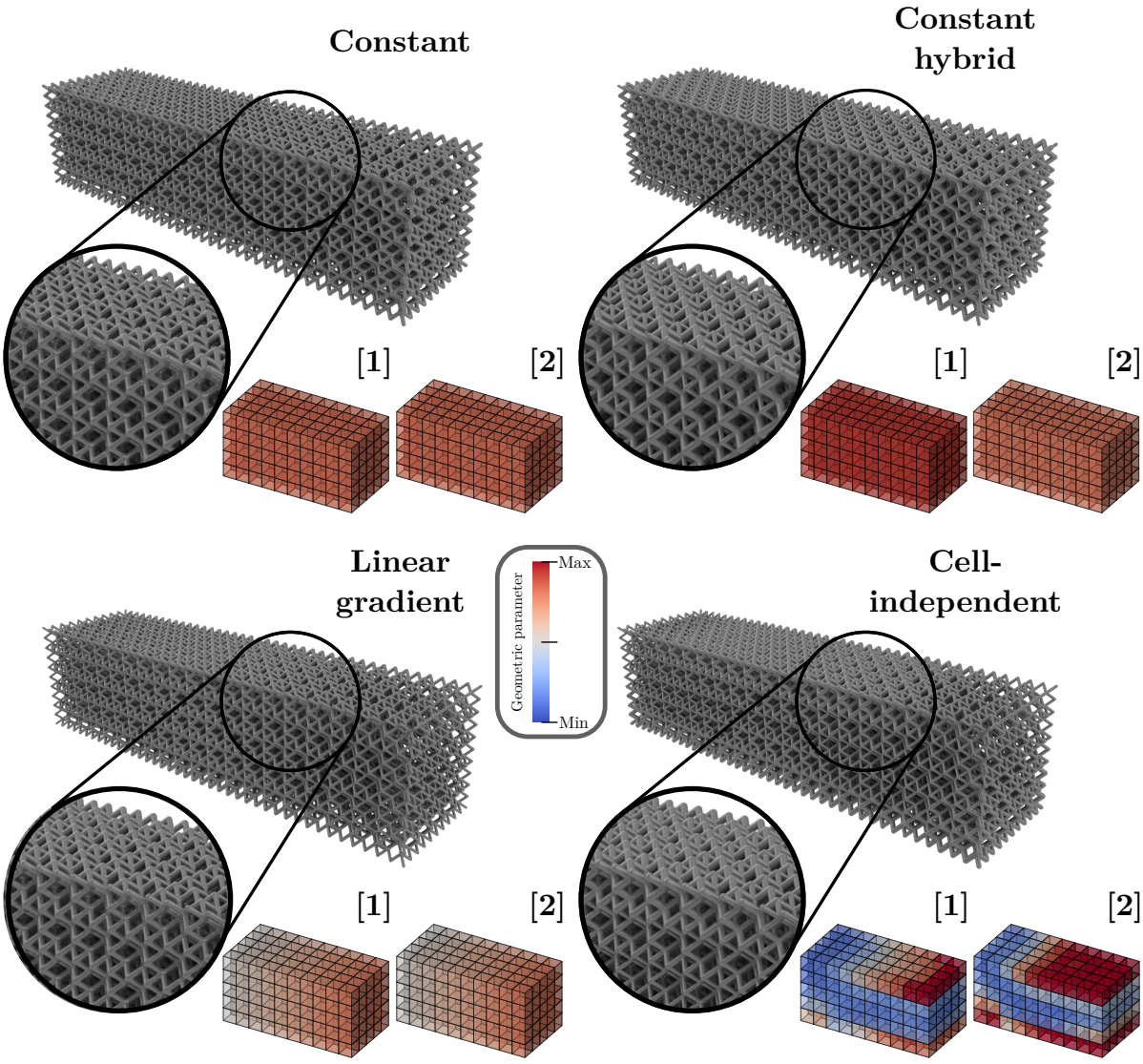

Fig. 5.4 Comparison of the optimization methods applied to a configuration composed of geometries [1, 2]. The colored voxel geometries illustrate the spatial arrangement of the different geometry types within each structure, with colors indicating the geometric parameter (here beam radius) from minimum (blue) to maximum (red).

across all geometry types. In contrast, the proposed method enables each cell to adapt independently, fully exploiting the strengths of each geometry where they are most effective, and thereby demonstrating the benefits of combining geometry hybridization with the optimization strategy.

The linear gradient method exhibits intermediate performance. While it allows some degree of variation in the design, it remains constrained by the global gradient direction. This method can be particularly useful when a specific grading direction is required, but

it may not fully capture the complexity of the loading conditions within the structure. Moreover, its effectiveness is highly dependent on the user-defined choice of gradient direction. Extending this approach to more complex gradient functions, such as higher-order polynomials or splines, or to multiple gradient directions, could provide additional flexibility in the design space. Nevertheless, such extensions would also increase the number of design variables and could complicate the optimization process.

Ranking and Discussion of Optimization Strategies

Moreover, Table 5.2 shows a ranked list of the compliance values obtained for all optimization methods and geometry combinations. The results clearly show that the proposed approach is most effective when combined with the hybrid cell geometry parameterization, with geometry 3 providing the best performing basis in this case. The linear gradient method also yields competitive results, particularly when applied to 3 alone or to combinations involving 3. Interestingly, the constant method reaches the 6th position with the hybrid cell geometry parameterization $[2, 3]_{\text{hyb}}$, which is a remarkable outcome given the simplicity of this approach, further demonstrating the efficiency of the hybridization strategy. A more comprehensive analysis involving multiple study cases would be required to establish broader conclusions regarding the relative performance of these methods under varying scenarios.

Overall, the results highlight the effectiveness of the proposed optimization strategy, which combines flexibility and computational efficiency, thereby providing a promising approach for the design of large-scale lattice structures. The findings also emphasize the crucial role of hybrid geometries, as well as the benefits of allowing independent parameter variations at the cell level to achieve superior structural performance. The choice of cell geometries used in the hybridization process has a decisive influence on the optimization outcome, and must therefore be carefully selected to fully exploit the potential of the method.

5.2.4 Case Study: Full Optimization Pipeline Applied to a Femoral Head Implant

In real-world applications, the design space can become highly complex, requiring additional processing steps. Figure 5.5 illustrates the different stages of the method for a femoral head implant (STL file obtained from Leiden (2015)). This example was selected because of its geometric complexity, the presence of multiple loading conditions that

Table 5.2 Optimization methods ranked in ascending order of compliance.

Rank	Method	Geometries	Compliance ($\times 10^{-3}$)	Improvement over Constant [1]
1	Cell-independent	$[1, 2, 3]_{\text{hyb}}$	1.043	84.78 %
2	Cell-independent	$[2, 3]_{\text{hyb}}$	1.288	81.21 %
3	Cell-independent	$[1, 3]_{\text{hyb}}$	1.445	78.92 %
4	Cell-independent	$[3]_{\text{hyb}}$	1.997	70.86 %
5	Linear gradient	[3]	2.225	67.54 %
6	Constant	$[2, 3]_{\text{hyb}}$	2.252	67.14 %
7	Linear gradient	[2, 3]	2.274	66.82 %
8	Constant	[3]	2.363	65.52 %
9	Linear gradient	[1, 3]	2.415	64.77 %
10	Cell-independent	$[1, 2]_{\text{hyb}}$	2.618	61.8 %
11	Constant	[2, 3]	2.683	60.85 %
12	Constant	[1, 3]	2.985	56.45 %
13	Linear gradient	[1, 2, 3]	3.096	54.83 %
14	Constant	$[1, 3]_{\text{hyb}}$	3.158	53.92 %
15	Constant	$[1, 2, 3]_{\text{hyb}}$	3.363	50.93 %
16	Constant	[1, 2, 3]	4.229	38.3 %
17	Linear gradient	[1, 2]	4.265	37.77 %
18	Constant	$[1, 2]_{\text{hyb}}$	4.486	34.55 %
19	Cell-independent	$[2]_{\text{hyb}}$	4.560	33.47 %
20	Linear gradient	[2]	4.743	30.8 %
21	Constant	[2]	5.019	26.77 %
22	Constant	[1, 2]	5.371	21.64 %
23	Constant	[1]	6.854	0 %
=	Linear gradient	[1]	6.854	0 %
=	Cell-independent	$[1]_{\text{hyb}}$	6.854	0 %

favor the efficient use of hybrid geometries, and the suitability of the design for AM, as the implant must be tailored to the patient's anatomy (Naceur et al. 2015). The method to optimize multifunctional lattice structures is composed of six main steps:

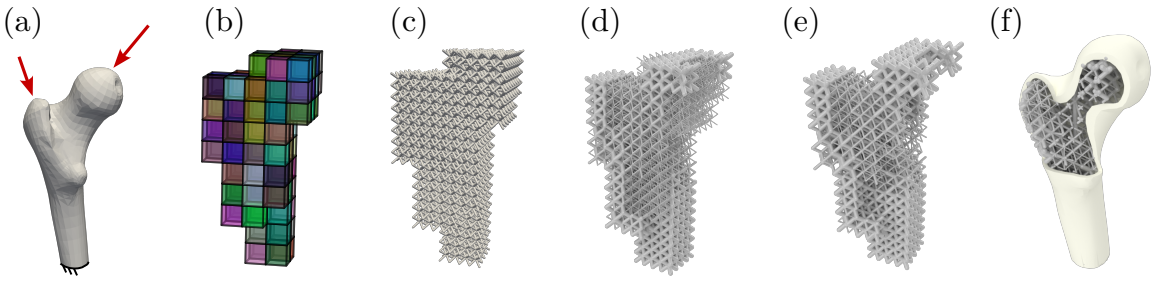

Fig. 5.5 Overview of the proposed method applied to a femoral head implant, including: (a) application of boundary conditions, (b) structure paving into cells, (c) cell population with lattice geometries, (d) structural optimization, (e) design post-processing, and (f) lattice reshaping to fit the initial geometry.

1. **Apply boundary conditions (a):** Impose the boundary conditions on the full-scale mesh of the structure.

2. **Pave the structure (b):** Discretize the structure into cells of size and dimension adapted to the geometry. Several strategies can be employed to pave complex geometries. In this work, we rely on a standard voxelization, in which the voxel grid slightly extends beyond the boundaries of the object to ensure complete coverage of the geometry. However, more advanced approaches with variable sizes or shapes of cells could also be considered. The only requirement is that a surrogate model is available to predict the behavior of each cell.

3. **Populate the cells (c):** Fill each cell with a lattice geometry, which may be beam-, surface-, or solid-based. Again, the constraint is to have a surrogate model capable of predicting the mechanical response of each chosen cell geometry.

4. **Optimize the structure (d):** Perform the optimization procedure described in Section 5.1.

5. **Post-process the design (e):** Apply post-processing operations to ensure manufacturability and structural coherence, as detailed in Section 5.2.2.

6. **Reshape the lattice (f):** Fit the lattice to the initial complex geometry. The simplest approach consists in trimming the lattice with the original mesh, but more advanced morphing or deformation techniques can also be applied. Ideally, morphing should be performed before optimization to better preserve physical consistency. In this work, the initial mesh is thickened to create a skin around the

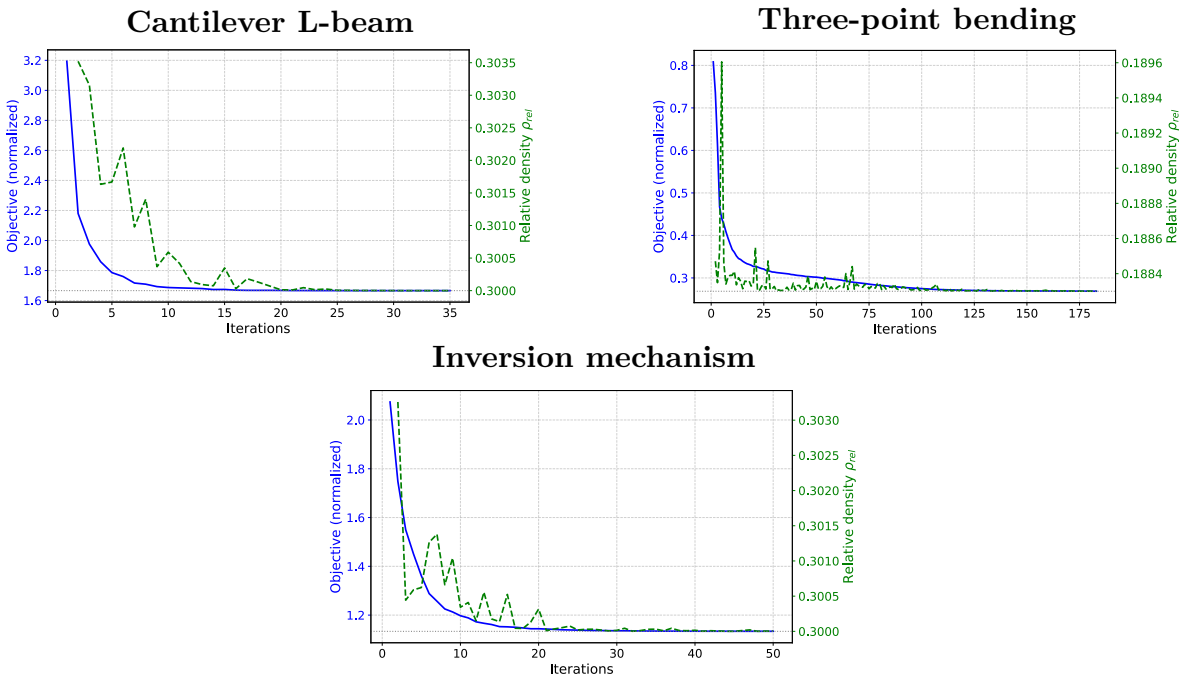

Fig. 5.6 Convergence curves for the three benchmark problems.

structure, although this step is not accounted for in the preceding stages of the method.

This pipeline provides a comprehensive framework for the optimization of multi-functional lattice structures within complex geometries. By combining domain decomposition with surrogate modeling, the method efficiently handles a large number of design variables, thereby ensuring its suitability for real-world applications.

5.3 Experimental Validation on Benchmark Structures

To validate the proposed optimization framework, the three benchmark problems introduced in Section 4.1.2 are analyzed. First, the study was conducted numerically, and subsequently, the specimens were additively manufactured for experimental testing.

5.3.1 Numerical Optimization Performance

Each benchmark problem is optimized using the cell-independent method with a hybrid configuration of the three base geometries. The optimization results for the different cases

Table 5.3 Optimization results for benchmark problems.

Case	ρ_{rel}	**No. param.**	**No. iter.**	**Opt. time (s) without gradient**	**Opt. time (s)**	**Compliance (Ref/Optim)** ($\times 10^{-3}$)	**Relative improvement**
Cantilever L-beam	0.3	81	35	9,847	325	Ref: 11.452 Optim: 4.403	+61.6 %
Three-point bending	0.1883	750	183	*N/A*	11,675	Ref: 2.897 Optim: 0.778	+73.1 %
Inversion mechanism	0.3	150	50	47,703	799	Ref: 2.527 Optim: 0.554	+78.0 %

are summarized in Table 5.3. A detailed visualization of the corresponding 3D lattice structures, before and after optimization, is provided later in the manuscript in Figure 5.7. The table reports the number of design parameters, the number of iterations, the total optimization time, and the compliance values for both the reference and optimized designs, along with the relative improvement. In addition, the convergence histories of the objective and relative density are presented in Figure 5.6, highlighting the rapid and stable convergence behavior of the proposed optimization framework. All cases have been optimized to minimize compliance under a prescribed volume constraint. Initially, a different optimization strategy was planned for the inversion mechanism, which consisted in maximizing the inverse of the output displacement at the end opposite to the applied load. Such an approach typically leads to the synthesis of a compliant mechanism that reverses the input displacement, as commonly proposed in the TO literature (Seltmann and Hasse 2023; D. Zhao and H. Wang 2023). However, due to time limitations in developing the optimization framework, a compliance minimization formulation was adopted instead. Overall, the simulation results demonstrate significant enhancements in mechanical performance across all study cases, with relative gains ranging from 61.6% to 78.0%. On the Cantilever L-beam benchmark, the optimization completes in less than 325 seconds when the sensitivity of each geometric parameter is directly computed, while it requires 9,847 seconds when using finite differences, demonstrating the remarkable computational efficiency of the proposed approach. It should be noted that all simulations are performed using the DDM framework; employing a standard FEM solver would lead to an even greater difference in computational time, as evidenced by the results reported in Table 4.2. Moreover, the benchmark studies highlight the strong correlation between

the number of design parameters and the computational cost. Without the proposed method, the optimization of lattice structures composed of a large number of cells would be computationally intractable.

5.3.2 Fabrication, Post-Processing, and Experimental Preparation

The three benchmark problems are fabricated using AM and subsequently subjected to experimental testing. The lattice specimens were fabricated using a Stratasys Objet260 Connex3 3D printer (Stratasys®, USA) through a Material Jetting (PolyJet™) AM process. The printing material was VeroWhite™ photopolymer, with a nozzle diameter of 30 μm, a layer thickness of 30 μm, and a printing resolution of 600 dpi. Mechanical testing was performed on a Lloyd LR 50K testing machine equipped with a 50 kN load cell, under displacement control, using a crosshead speed of 2 mm/min. Each benchmark case is printed in two configurations: a reference design with a uniform lattice distribution, and an optimized design obtained using the proposed method with the same relative density. The characteristics of the printed specimens are summarized in Table 5.4. The specimen dimensions are selected to ensure compatibility with the available testing equipment. The three-point bending beam can be tested directly without any modification, since the loading and boundary conditions are applicable to the printed specimen. However, some adaptations are required for the Cantilever L-beam and the Inversion mechanism to ensure the testability of the specimens on the available testing machines. For the Cantilever L-beam, a block is added at the clamped end to facilitate gripping in the machine, and a cylindrical connector is introduced between the load application point and the loading head to ensure proper load transfer. For the Inversion mechanism, two blocks are attached at the clamped ends to allow reliable gripping of the specimen. An overview of the printed specimens and their testing configurations is shown in Figure 5.7. After printing, a post-processing procedure was performed to remove the support material. This step is essential to ensure the mechanical integrity of the specimens, as any residual support material could affect the accuracy of the experimental results. The post-processing consisted of two stages. First, the external support material was removed using high-pressure water jets. Then, the specimens were immersed in a sodium hydroxide aqueous bath to dissolve any remaining support material and to ensure that all internal cavities were completely cleared. Finally, the specimens were thoroughly cleaned before testing. However, it is important to note that

the post-processing procedure may not be optimal for the complex lattice structures considered in this study. The removal of support material from intricate geometries can be challenging, and there is a risk of damaging delicate features during the cleaning process.

Table 5.4 Description of the printed specimens.

Case	**Cell dim. (mm^3)**	**Printing time (h)**	**Weight (g)**	**Adaption for testing**
Cantilever L-beam	20	10	Ref: 163.33 Optim: 166.45	Yes Weight: 90.9g
Three-point bending	10	6.5	Ref: 98.67 Optim: 105.11	No
Inversion mechanism	15	5	Ref: 130.30 Optim: 150.15	Yes Weight: 21.5g

Remark. The pipeline from the optimized design to the printed specimen is not straightforward, as several steps remain challenging. First, the meshing process is difficult due to the coexistence of multiple scales and very thin features. In this work, the mesh was generated uniformly over the entire structure, for simplicity reason. However, capturing the thin beams requires a very fine mesh, which leads to an extremely large number of elements and files exceeding 1 GB in some cases. Future improvements could rely on advanced meshing strategies with local refinement, or alternatively on methods allowing a direct conversion from the beam model to the printing file. Second, material jetting technology offers high resolution and great design freedom, making it a convenient choice for manufacturing lattice structures. Nevertheless, this technique requires the use of support material throughout the part, which generates substantial waste and demands long post-processing times to remove it. In practice, the mass of the support material is often greater than that of the printed structure itself. Further developments in support-free printing techniques are required to fully exploit the potential of AM for complex lattice structures.

Figure 5.7 presents the reference and optimized specimens for each benchmark case. The spatial distribution of the different cell geometries within each structure is also shown, with colors indicating the geometric parameter from minimum (blue) to maximum (red). This visualization provides insight into how the optimization process tailored the lattice structures to the specific loading conditions and constraints of each case by efficiently exploiting the available design space. A key observation is that each geometry type is

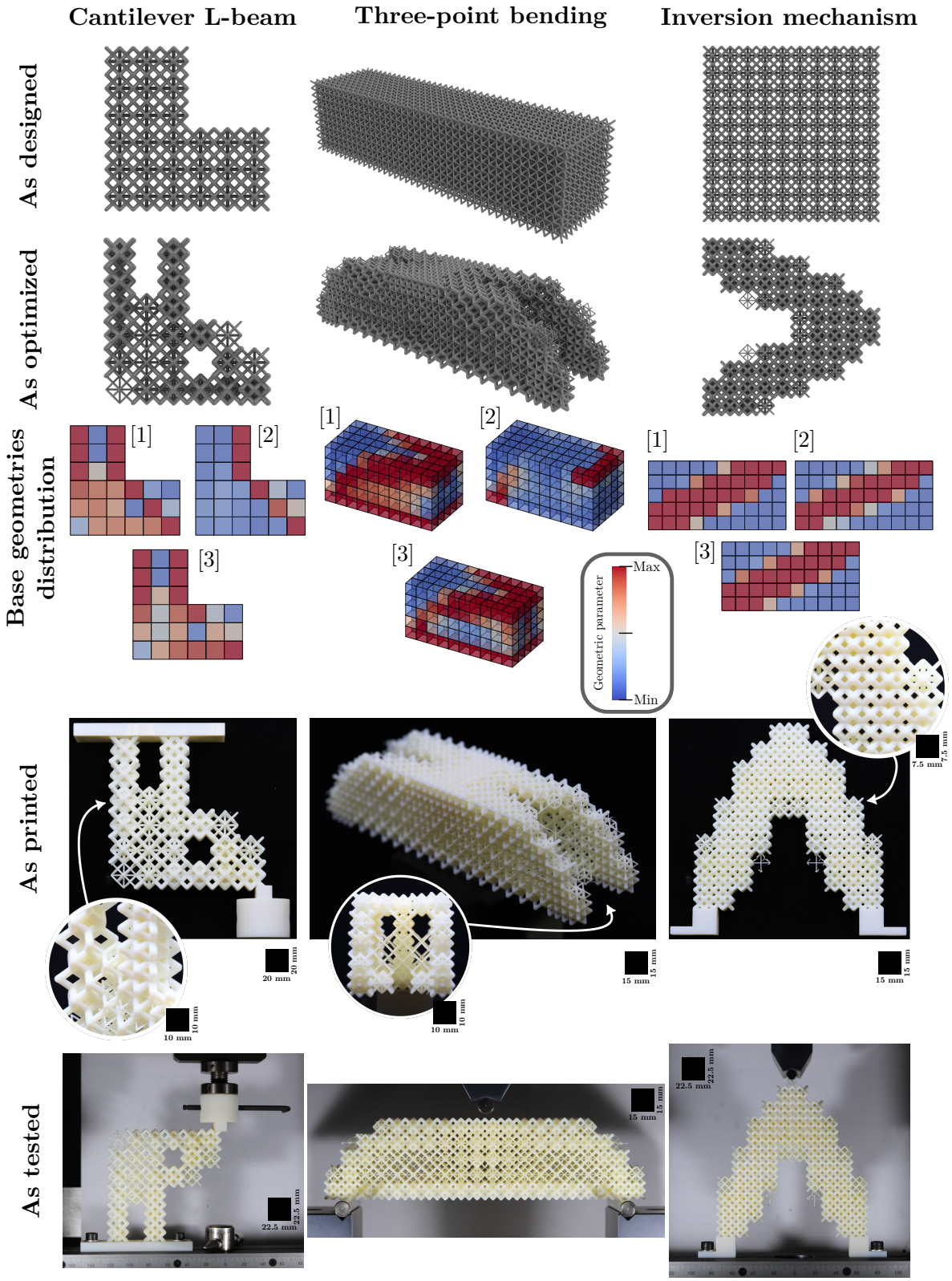

Fig. 5.7 Reference and optimized benchmark cases investigated in this study. The base geometries distribution row illustrates the spatial arrangement of the different geometry types within each structure, with colors indicating the geometric parameter (here beam radius) from minimum (blue) to maximum (red). The bottom row shows the specimens as printed with the material jetting process, demonstrating both the feasibility of manufacturing the optimized lattice structures and the adaptations introduced to enable mechanical testing.

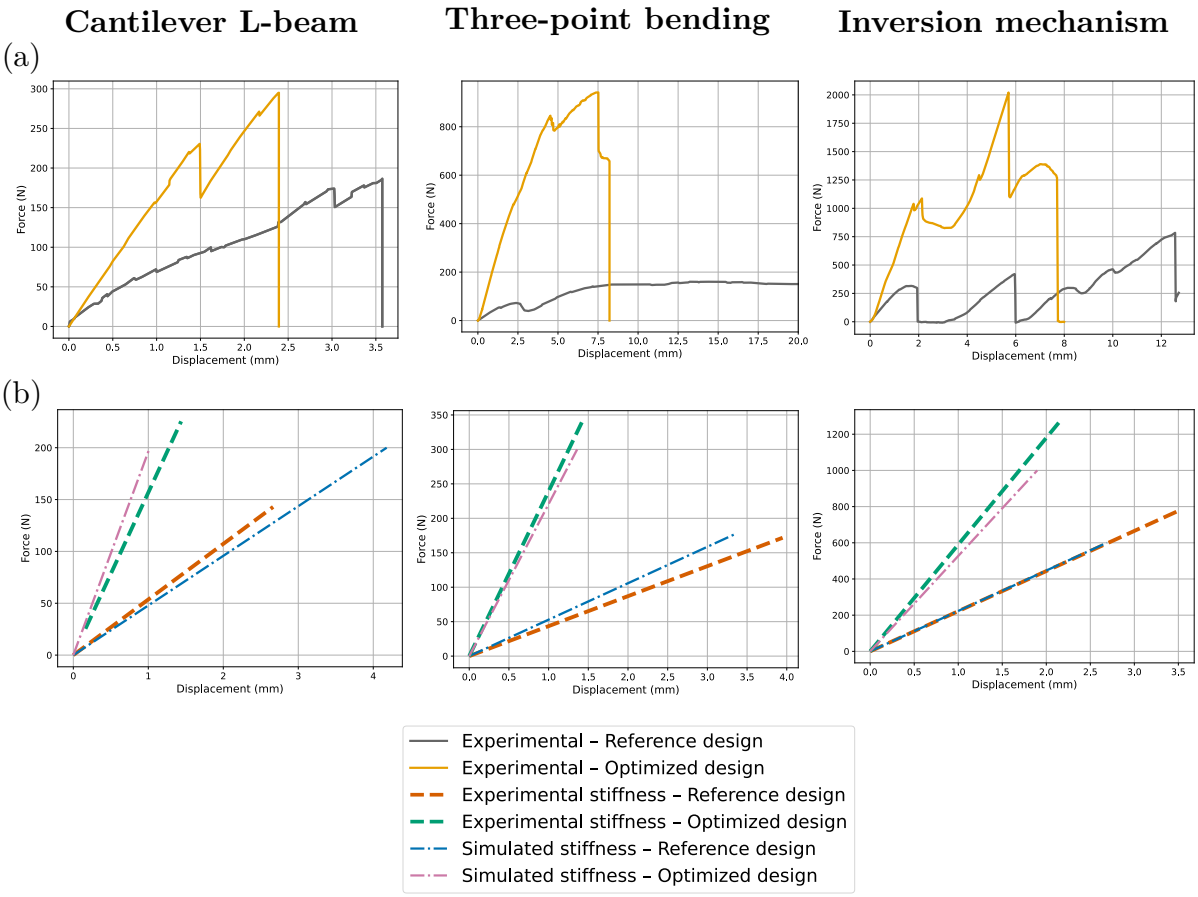

Fig. 5.8 Force–displacement curves obtained from experimental testing and numerical simulations for the three benchmark cases. (a) Unprocessed experimental results, (b) Stiffness comparison between experimental measurements and numerical predictions.

allocated to different regions of the structure depending on its mechanical performance. For instance, in the Cantilever L-beam, geometry 2 appears only on the right side of the fixed end, where it is the most effective. In the Three-point bending beam, geometries 1 and 3 are widely distributed throughout the structure, while geometry 2 is restricted close to the boundary conditions. In contrast, the Inversion mechanism employs the three geometries in nearly identical proportions and at similar locations across the structure. This highlights the crucial role of base geometry selection within the design space, as it strongly influences the optimization outcome.

5.3.3 Experimental Results and Discussion

Figure 5.8 presents the force–displacement curves obtained from experimental testing and numerical simulations for the three benchmark cases. Moreover, Table 5.5 summarizes

Case	Stiffness (N/mm)	Simulation relative error (%)	Stiffness relative to mass	Stiffness improvement (%)
Cantilever L-beam	Expe. ref.: 53.6 Simu. ref.: 47.9 Expe. opti.: 156.8 Simu. opti.: 196.2	ref.: 10.6% opti.: 25.1%	ref.: 0.74 opti.: 2.07	192%
Three-point bending	Expe. ref.: 43.5 Simu. ref.: 52.9 Expe. opti.: 239.2 Simu. opti.: 217.8	ref.: 21.6% opti.: 8.9%	ref.: 0.46 opti.: 2.28	449%
Inversion mechanism	Expe. ref.: 221.8 Simu. ref.: 223.3 Expe. opti.: 590.0 Simu. opti.: 527.1	ref.: 0.6% opti.: 10.6%	ref.: 2.03 opti.: 4.58	166%

Table 5.5 Comparison between experimental and simulated stiffness values for the reference and optimized designs of each benchmark case. *Ref.* and *Opti.* refer to the reference and optimized designs, respectively, while *Expe.* and *Simu.* denote the experimental and simulation results.

the stiffness values extracted from both experimental and simulated results, along with the relative errors between the two approaches. Videos of the experimental tests are also available online for reference.[2] In the unprocessed results, the shift in behavior from brittle to ductile can be observed between the reference and optimized designs, particularly in the three-point bending case. In our framework, only the linear elastic regime is considered. For the sake of comparison, the stiffness is evaluated as the slope of the linear segment of the force–displacement response, measured within the first millimeter of displacement. The results demonstrate a good agreement between the experimental measurements and the numerical predictions, with relative errors ranging from 0.6% to 25.1%. The errors can be attributed to several factors, including manufacturing imperfections, the destruction of thin features during post-processing, and the incomplete removal of support material from internal cavities. The effect of incomplete removal of support material is particularly significant in optimized designs, where the presence of thicker beams and smaller cavities can hinder the cleaning process. Overall, the experimental results confirm the trends observed in the simulations, with the optimized designs exhibiting

[2]Video recordings of the experiments are available at: https://youtu.be/TlEw_J3dU0o

significantly improved mechanical performance compared to the reference designs. The optimization led to stiffness improvements of up to 166% for the Inversion mechanism, 192% for the Cantilever L-beam and even 449% for the Three-point bending. These results demonstrate the effectiveness of the proposed optimization framework in enhancing the structural performance of lattice structures.

5.4 Concluding Remarks on the Optimization of Locally Tailored Lattice Structures

This chapter presents a novel optimization framework for locally tailored lattice structures, combining domain decomposition and surrogate modeling to efficiently handle gradient computations. Through a series of numerical experiments, the method demonstrates its capability to optimize complex lattice architectures effectively. The integration of multiple base geometries within each cell proves particularly advantageous, as it enables the optimization process to exploit the specific strengths of each geometry type in different regions of the structure. Post-processing steps are introduced to ensure both printability and structural coherence of the optimized designs. The framework is further validated through experimental testing on benchmark structures, confirming the significant improvements in mechanical performance achieved through optimization. Moreover, the experimental results show good agreement with numerical predictions, validating the accuracy of the proposed simulation method and the reliability of the overall optimization procedure. The proposed approach provides a promising avenue for the design of multifunctional lattice structures, with potential applications across various engineering domains, as demonstrated by the complete optimization pipeline applied to a femoral head implant in Section 5.2.4.

Chapter 6

Conclusion And Perspectives

"Beyond the mountains, more mountains." — Haitian proverb

6.1 Final Overview

The core objective of this thesis was to develop an optimization method for lattice structures that is both efficient and flexible. Lattice structures are increasingly used in various engineering fields due to their exceptional mechanical properties, yet their design and optimization remain challenging because of their geometric complexity and inherent multiscale nature, which often hinder the direct application of conventional simulation and optimization methods.

Building upon this motivation, Chapter 3 addressed the simulation of lattice structures, the simulation of truss-based lattice structures modeled with beam elements. Several strategies were examined to compensate for the lack of stiffness at joints, revealing that existing approaches are often case-dependent and not easily generalizable. Through a series of numerical investigations, we proposed an improved beam penalization method that combines the strengths of existing techniques while extending their applicability to a wider range of lattice configurations. The method was further extended to dynamic simulations and non-linear material behaviors, with experimental validations confirming its robustness and computational efficiency. This foundation was essential to ensure accurate yet computationally efficient simulations of lattice structures.

In Chapter 4, the simulation framework was extended by coupling domain decomposition with surrogate modeling. The natural cell-based decomposition of

lattice structures enables a straightforward integration into the domain decomposition framework. Although the iterative coupling between local and global problems is computationally demanding, the introduction of surrogate models that efficiently approximate local behaviors at reduced cost significantly accelerates the global process. Two distinct surrogate modeling strategies were introduced, the pROM and the PANN, each learning the relationship between geometric parameters and mechanical responses in a different manner. Their respective performance and limitations were analyzed to identify the most suitable approach for complex lattice structures.

Finally, Chapter 5 integrated the developed simulation methodology into an optimization framework dedicated to the design of optimal lattice structures. A parametrization approach was introduced, in which each cell is defined by several geometric parameters, making the total number of design variables proportional to the number of cells in the structure. In this context, the developed surrogated domain decomposition method provides a dual advantage: it accelerates the structural analysis and directly delivers the sensitivities with respect to the geometric parameters of each cell. This capability enables the efficient application of gradient-based optimization algorithms to high-dimensional design problems. The performance of the proposed framework was demonstrated through a series of numerical experiments and experimental validations on benchmark cases, confirming its robustness, accuracy, and practical efficiency.

Overall, this thesis presents a unified and computationally efficient framework that couples an advanced simulation approach with the design flexibility of lattice structures. The resulting capability for mass-efficient designs makes the approach particularly attractive for lightweight engineering applications, such as transportation where energy efficiency is a critical objective.

However, this study represents a first step toward the development of a global methodology, and several limitations still remain. The main limitation lies in the surrogate modeling stage. To progress further, a more accurate and better-performing model is required to reliably predict the mechanical behavior of lattice structures. In addition, the domain decomposition algorithm could be extended to accommodate alternative modeling approaches beyond beam elements, such as shell or even 3D solid formulations. Further research on the optimization stage is also required to conduct a more in-depth assessment of the current algorithm and to explore more advanced strategies capable of

improving convergence efficiency. Finally, extending the optimization problem to include more complex constraints would allow the framework to address advanced scenarios, such as better managing geometric connectivity.

6.2 Research Perspectives

Although this thesis has addressed several key questions, many challenges and open issues remain to be explored. Each aspect presented in this section is positioned according to its expected maturity and the time horizon required to obtain meaningful answers.

6.2.1 Short-term perspectives

The developed optimization method has demonstrated excellent performance across the three benchmark studies. However, numerous possibilities remain to be explored in order to fully exploit the potential of the proposed approach. Focusing on beam-based lattice structures, alternative cell parameterizations could be readily developed to enrich the design space. For instance, the current hybrid parameterization could be extended by incorporating point-based descriptors or curved beam geometries, or even a combination of beam and surface modeling, enabling smoother and more versatile structural variations. In addition, the use of multi-material configurations, made possible by material jetting additive manufacturing technologies, represents a promising direction. Such extensions require no major modification of the proposed framework and can be adapted to a wide range of applications, from medical devices such as customized insoles to large-scale engineering components.

The transition from optimal designs composed of one-dimensional beam elements to printed specimens has revealed several difficulties, mainly related to meshing process, and printer capabilities. A more direct link to the printed specimen could be established through implicit toolpath generation (D. Adams and Turner 2018; Wade et al. 2025), thereby eliminating the need to mesh the entire structure. Furthermore, integrating the printing machine capabilities and post-processing constraints within the optimization algorithm would enhance the generality of the proposed method, making it applicable to a wider range of manufacturing processes. For example, considering factors such as minimum printable size, the severity of post-processing conditions, and maximum overhang angle can minimize the differences between simulation and experiment.

6.2.2 Medium-term perspectives

Linear elasticity provides a solid foundation for the development of a simulation framework. However, modeling more advanced material and structural behaviors would open the door to a wider range of applications. The proposed surrogate model based on the PANN shows strong potential for extension to dynamic analyses, where increased physical complexity can be efficiently handled by NN capable of capturing nonlinear relationships. As discussed in Section 4.4.4, the current surrogate model still lacks the accuracy required to fully reproduce the structural response. Future work could therefore focus on implementing more advanced NN architectures while maintaining the physical integration within the learning process to ensure both efficiency and consistency of the implementation. In particular, GNN appear as a promising direction, since the geometry of a lattice cell can be naturally represented as a graph, offering a more natural and physically meaningful description than traditional NN architectures. Preliminary attempts in this direction were conducted during this research but were limited by the complexity of the implementation. Nevertheless, extending the simulation capabilities beyond linear elasticity could greatly broaden the range of potential applications, such as modeling energy absorption mechanisms in protective equipment or other nonlinear phenomena.

6.2.3 Long-term perspectives

The ultimate goal of the proposed framework is to efficiently handle the optimal design of structures with complex and high-dimensional design spaces, potentially involving fully parameterized 3D cell geometries composed of multiple materials. Moreover, extending the framework toward compatibility with multi-physics lattice structures—such as coupled fluid–thermal–mechanical systems—would be highly valuable, as such lattices hold great promise for advanced applications, including heat exchangers and thermal management systems.

Chapter 7

Résumé des travaux en Français

"The use of English as the common language of science represents a major impediment to maximising the contribution of non-native English speakers to science." —(Amano et al. 2023)

Contents

7.1 Introduction 182
7.2 Rapide présentation de l'état de l'art et problématique 183
7.2.1 Structures lattices et fabrication additive 183
7.2.2 Simulation des structures lattices 184
7.2.3 Optimisation des structures lattices 185
7.2.4 Verrous scientifiques et positionnement de la démarche 186
7.3 Méthode de pénalisation aux nœuds 187
7.3.1 Contexte et motivation 187
7.3.2 Méthodologie 188
7.3.3 Extension aux cas non linéaires et validation expérimentale 194
7.3.4 Conclusion 194
7.4 Décomposition de domaine pour les structures lattices 197
7.4.1 Fondations pour l'exploration des lattices localement optimisées . . . 197
7.4.2 Méthode de décomposition de domaine appliquée aux structures lattices 197
7.4.3 Modèle de substitution pour l'analyse rapide 201
7.4.4 Modèle de substitution augmenté par la physique 204
7.4.5 Discussion et conclusion 206
7.5 Vers une optimisation locale efficace des structures lattices . . . 206
7.5.1 Étude de cas : Prothèse de tête fémorale 207
7.5.2 Validation expérimentale sur les structures de référence 208
7.6 pyLattice : un outil de conception, de simulation et d'optimisation 210
7.7 Conclusion et perspectives 212

7.1 Introduction

L'émergence de la fabrication additive a profondément transformé les possibilités en matière de conception, en permettant la réalisation de structures aux géométries complexes, auparavant inaccessibles. Parmi ces structures, les structures lattices (Seharing et al. 2020; Y. Wu, Fang, et al. 2023) se distinguent par leur capacité à concilier légèreté, performances mécaniques accrues et propriétés multiphysiques personnalisables. Elles offrent un potentiel considérable dans les domaines de la mobilité (Opgenoord and Willcox 2018; *X VEIN* 2020), de la santé (*Lattice Medical* 2023; *Orthopedic Implants AM* 2024) ou de la protection (*Riddell Diamond Technology* 2019), en permettant par exemple de réduire la masse des pièces, d'absorber l'énergie d'impact ou encore de moduler localement les propriétés mécaniques. Cette liberté de conception s'accompagne cependant d'une complexification importante du processus de simulation et d'optimisation, en raison de la richesse de l'espace de conception et de la finesse des détails géométriques à prendre en compte.

Pour pallier ces difficultés, de nouvelles méthodes numériques sont nécessaires afin de réduire le coût de calcul associé à l'analyse et à l'optimisation de telles structures, tout en conservant une précision suffisante. L'objectif de cette thèse est donc de proposer une approche performante permettant de concevoir des structures lattices optimisées pour des applications spécifiques, en tenant compte des contraintes numériques liées à leur complexité géométrique. Pour cela, nous avons d'abord développé une méthode efficace de simulation du comportement mécanique des structures lattices en nous appuyant sur des éléments poutres enrichis de pénalisations au niveau des jointures. Cette modélisation, plus rapide que les approches éléments finis 3D classiques, permet une évaluation fiable du comportement structural tout en réduisant significativement les temps de calcul. À partir de cette base, nous avons mis en place une chaîne d'optimisation modulaire capable de traiter différentes géométries de cellules unitaires, différents paramètres de conception, ainsi que divers objectifs fonctionnels à l'échelle macroscopique. Cette méthode repose sur une flexibilité suffisante pour intégrer des modèles multi-physiques, multi-échelles et multi-matériaux, dans le but de concevoir des structures optimales aux comportements complexes.

7.2 Rapide présentation de l'état de l'art et problématique

7.2.1 Structures lattices et fabrication additive

L'essor des technologies de Fabrication Additive (FA) a profondément transformé la manière dont les structures sont conçues et fabriquées, de nouvelles opportunités ont émergé, notamment les structures lattices (Seharing et al. 2020; Y. Wu, Fang, et al. 2023). Contrairement aux matériaux classiques, ces structures architecturées permettent de dissocier la composition matérielle de la réponse mécanique globale, ouvrant ainsi la voie à des comportements mécaniques inédits. En effet, les structures lattices sont constituées d'un réseau périodique de cellules unitaires, formant un agencement tridimensionnel de vides et de matière. Les structures lattices se distinguent par une grande diversité topologique, allant de réseaux 2D simples à des architectures 3D complexes, y compris des structures à base de poutres, de plaques ou de surfaces triplement périodiques (Voir Figure 7.1). Leur comportement dépend fortement de la topologie de la cellule unitaire, de la densité relative et du matériau constitutif, permettant de concevoir des structures aux propriétés mécaniques, thermiques, acoustiques ou vibratoires ciblées (Yin, W. Zhang, et al. 2023; Zhou et al. 2025). Des stratégies de gradation spatiale peuvent également être mises en œuvre, notamment via des gradients de densité, de taille de cellule ou de type géométrique, pour optimiser les performances fonctionnelles (Y. Li et al. 2020). La fabrication de ces structures repose majoritairement sur la FA, qui permet de réaliser des géométries complexes, inaccessibles par les procédés traditionnels. Plusieurs technologies de FA sont mobilisées selon les matériaux et les résolutions souhaitées : dépôt de fil fondu (FFF), photopolymérisation en cuve (SLA, DLP), fusion de lit de poudre (Selective Laser Melting (SLM), Electron Beam Melting (EBM)) et d'autres encore. Chaque procédé présente des avantages et des limites en termes de précision, de matériaux utilisables et de propriétés mécaniques des pièces produites. Les structures lattices trouvent aujourd'hui des applications dans de nombreux domaines, allant des semelles de chaussures personnalisées (*Adidas 4DFWD* 2021) aux implants biomédicaux (*Orthopedic Implants AM* 2024), en passant par les composants allégés pour l'aéronautique (Opgenoord and Willcox 2018) ou les systèmes d'absorption d'énergie dans les casques de protection (*Riddell Diamond Technology* 2019).

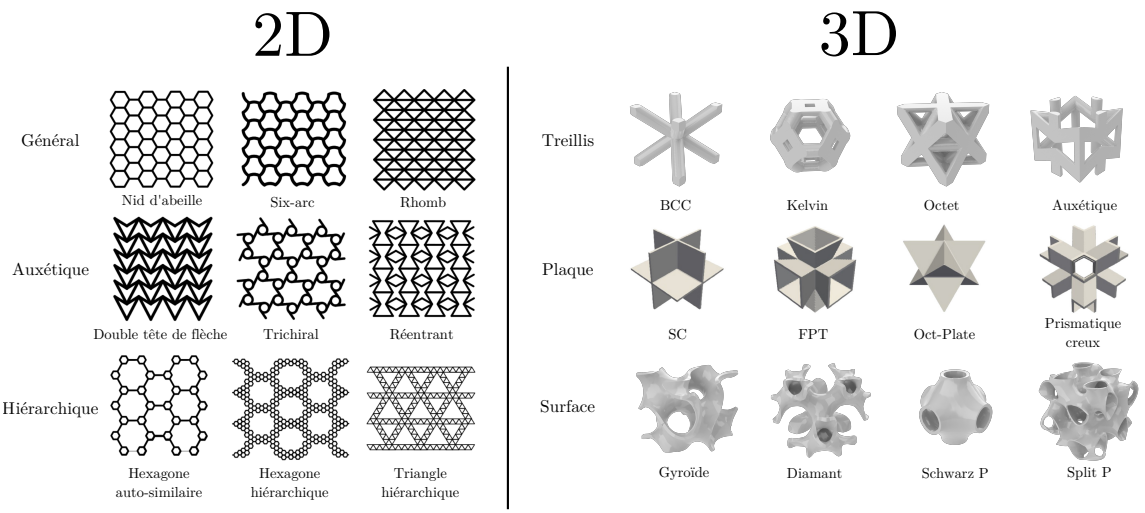

Fig. 7.1 Aperçu de la variété des structures lattices. (Structure 2D adaptée avec permission depuis (Yin, W. Zhang, et al. 2023), structure à plaques adaptée avec permission depuis (B. Sun et al. 2023)).

7.2.2 Simulation des structures lattices

L'évaluation précise et efficace des propriétés des structures lattices est un enjeu majeur pour orienter leur conception et optimiser leurs performances. Différentes approches existent, du modèle analytique aux simulations numériques avancées, en passant par la caractérisation expérimentale. Parmi les stratégies numériques, les méthodes à une seule échelle (mono-échelle) permettent de simuler directement la structure complète, via des éléments finis 3D ou des modèles simplifiés tels que poutres ou coques. Les modèles simplifiés permettent de réduire le coût de calcul, mais nécessitent des ajustements pour corriger les rigidités locales aux nœuds, notamment pour les éléments poutres. Les approches multi-échelles, quant à elles, reposent sur des techniques d'homogénéisation ou de décomposition de domaine. L'homogénéisation, fondée sur une séparation claire des échelles, permet de calculer un comportement effectif à partir de cellules unitaires représentatives, mais devient imprécise en cas de gradient ou d'hétérogénéité des cellules. À l'inverse, la décomposition de domaine, ne nécessitant pas de séparation d'échelle, permet de simuler des structures complexes en résolvant des sous-problèmes locaux couplés par des conditions d'interface. Enfin, la caractérisation expérimentale reste incontournable pour valider les modèles numériques, en particulier dans le cas de géométries complexes ou de comportements non linéaires. Les essais de compression, de traction ou d'impact sont utilisés pour évaluer les performances mécaniques, dynamiques ou d'absorption d'énergie des structures lattices. Cependant, les écarts entre simulation et réalité peuvent

provenir de défauts de fabrication liés aux procédés de FA, tels que la porosité ou les imprécisions géométriques. En conclusion, la simulation des structures lattices repose sur un compromis entre précision, coût de calcul et faisabilité numérique. Une stratégie rigoureuse combine souvent modèles réduits, simulations haute-fidélité et validation expérimentale pour guider la conception de structures performantes.

7.2.3 Optimisation des structures lattices

L'optimisation des structures lattices constitue une problématique complexe en raison de leur nature multi-échelle et du nombre élevé de paramètres à considérer. Trois grandes stratégies sont classiquement distinguées : l'optimisation dimensionnelle (modification de paramètres géométriques prédéfinis), l'optimisation de forme (adaptation des contours de la structure), et l'optimisation topologique (répartition optimale de matière dans un domaine donné). Dans le cas des lattices, ces approches peuvent être appliquées à différentes échelles :

À l'échelle de la cellule unitaire, les travaux portent sur l'optimisation de la densité relative, de la géométrie interne ou de la répartition multimatériau, afin d'améliorer des critères tels que la raideur ou la capacité d'absorption d'énergie.

À l'échelle macroscopique, les stratégies visent à adapter la répartition spatiale des cellules unitaires ou à concevoir des structures conformes aux contraintes géométriques et fonctionnelles globales. Les méthodes basées sur l'homogénéisation sont largement utilisées pour intégrer les propriétés effectives des cellules dans un cadre d'optimisation topologique. Cependant, l'hypothèse de séparation d'échelle inhérente à l'homogénéisation est souvent violée dans les structures avec gradients, ce qui limite sa validité.

Les approches multi-échelles permettent d'optimiser simultanément la géométrie des cellules et leur répartition au sein de la structure. Cela peut se faire par des modèles géométriques simplifiés, des méthodes itératives alternant entre micro et macro-échelles, ou des pipelines intégrés basés sur des réseaux neuronaux. Ces méthodes offrent de grandes possibilités, mais posent des défis en termes de coût computationnel et de complexité d'implémentation.

Les métaheuristiques, comme les algorithmes génétiques (GA), l'optimisation par essaim particulaire (PSO) ou le recuit simulé(SA), offrent une alternative robuste aux méthodes basées sur le gradient, notamment pour les problèmes non convexes ou multi-objectifs. Leur capacité à explorer de vastes espaces de conception sans dépendre d'informations différentielles les rend particulièrement attractives dans le contexte des lattices, notamment lorsqu'elles sont combinées à des modèles de substitution. En

conclusion, le choix de la stratégie d'optimisation dépend fortement du niveau d'échelle considéré, des ressources disponibles, et des objectifs de performance. Une tendance actuelle consiste à combiner plusieurs approches dans un cadre unifié, tout en intégrant des modèles de substitution pour réduire le coût des évaluations. Cette approche intégrée est au cœur des développements présentés dans les sections suivantes.

7.2.4 Verrous scientifiques et positionnement de la démarche

La montée en complexité des structures lattices, rendue possible par les avancées de la fabrication additive, ouvre de nouvelles perspectives en matière de conception et d'optimisation. Cependant, il s'accompagne de verrous scientifiques tout au long de la chaîne numérique, depuis la modélisation jusqu'à la fabrication. Il devient nécessaire d'adopter une vision globale et intégrée du processus de conception, car une segmentation en étapes isolées compromet la cohérence et l'efficacité de l'approche. Un premier verrou concerne la capacité à générer des structures lattices avec une paramétrisation claire et contrôlable. Cette thèse se concentre sur les structures lattices de type treillis pour trois raisons principales : leurs excellentes performances mécaniques (absorption d'énergie, raideur spécifique), leur complexité atteignable avec peu de paramètres, et la possibilité d'évaluation rapide via la théorie des poutres. Cependant, la littérature montre que les hypothèses simplificatrices de la théorie des poutres limitent fortement la précision des prédictions mécaniques. Aucune méthode généralisée n'existe pour corriger ces approximations dans toutes les structures treillis. Le premier apport de cette thèse, présenté à la section 7.3, est le développement d'une méthode de pénalisation optimale permettant de corriger la raideur des nœuds. Une fonction géométrique est utilisée pour identifier les zones de jonction et corriger la quantité appropriée de matière selon chaque géométrie. Une fois cette méthode établie, un second verrou émerge : concevoir des structures avec des propriétés mécaniques ciblées et multifonctionnelles, notamment par l'agencement optimal de différentes cellules dans une même structure. Contrairement aux approches classiques qui utilisent une seule cellule optimisée, cette stratégie permet d'adapter localement les propriétés mécaniques. Cependant, elle soulève un défi computationnel majeur : l'évaluation coûteuse des propriétés globales lorsque chaque cellule possède ses propres paramètres. La seconde contribution de cette thèse, détaillée à la section 7.4, repose sur une méthode de décomposition de domaine couplée à un modèle de substitution de type réseau de neurones augmenté par la physique, permettant d'estimer efficacement les propriétés mécaniques d'une structure contenant plusieurs géométries de cellules. Cette approche ouvre de nouvelles perspectives en matière

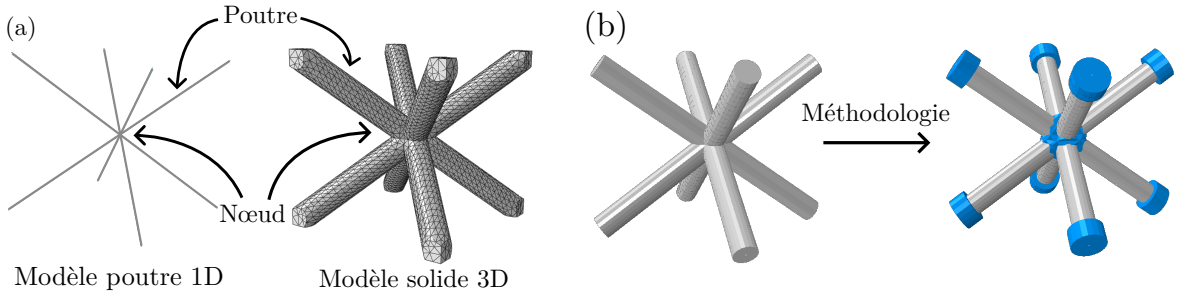

Fig. 7.2 (a) Comparaison des dimensions des nœuds selon le modèle poutre et le modèle 3D pour la cellule BCC. (b) Méthodologie appliquée à la cellule unitaire BCC.

de conception optimale de structures lattices complexes. Enfin, la méthode de simulation est intégrée dans un cadre d'optimisation de structures lattices multifonctionnelles, permettant la conception de structures adaptées à des applications spécifiques, comme présenté dans la section 7.5.

7.3 Méthode de modélisation par éléments poutres avec pénalisation aux nœuds

7.3.1 Contexte et motivation

Les structures lattices basées sur des poutres (modèles treillis) suscitent un intérêt croissant en raison de leurs excellentes propriétés mécaniques, telles qu'une anisotropie contrôlable et une capacité d'absorption d'énergie élevée. Dans un contexte numérique, leur modélisation repose souvent sur des modèles poutres 1D, nettement moins coûteux en temps de calcul et en ressources mémoire que les modèles volumiques 3D. Cela permet de simuler efficacement des structures de grande taille ou à géométrie complexe. Cependant, une limite majeure des modèles poutres réside dans leur incapacité à représenter fidèlement la rigidité des jonctions, c'est-à-dire les zones auxquelles plusieurs poutres se connectent. En effet, contrairement aux modèles 3D qui intègrent naturellement le volume et la masse des nœuds, les modèles poutres les représentent par un point infinitésimal, simplifiant ainsi leur comportement et négligeant leur contribution mécanique réelle (Voir Figure 7.2). Cette approximation entraîne une sous-estimation notable de la rigidité globale de la structure, surtout dans les cas de forte densité relative ou lorsque les nœuds jouent un rôle structurant essentiel, ce qui est le cas dans la majorité des structures lattices. De nombreuses approches ont été proposées dans la littérature pour compenser cette limitation, allant de l'augmentation locale du module d'Young, à l'élargissement de

la section des poutres aux extrémités, ou encore à l'utilisation de méthodes hybrides combinant éléments solides et poutres (Gärtner et al. 2021; H. Guo et al. 2020; Lozanovski, Downing, Tino, et al. 2020; Lozanovski, Downing, Tran, et al. 2020; L. Meng et al. 2020; Smith et al. 2013; Tahmasebimoradi et al. 2021; Weeger, Valizadeh, et al. 2023). Toutefois, aucune stratégie n'a encore fait consensus, en particulier pour des structures de densité variable, de géométrie arbitraire, ou pour la modélisation dynamique de leur comportement.

Objectif de l'étude

Cette section propose une méthodologie générique et non intrusive permettant d'améliorer la précision des modèles poutres dans la simulation des structures lattices. L'approche repose sur une pénalisation géométrique locale des jonctions, en augmentant la section des poutres dans les zones de connexion sur une longueur donnée, sans changer la formulation élément fini. Elle vise à corriger le déficit de rigidité au niveau des nœuds tout en conservant l'efficacité numérique des modèles poutres. Trois paramètres principaux sont introduits pour cette correction :

- l_{zone} : la longueur de jonction
- α_{r} : un coefficient de pénalisation du rayon permettant d'augmenter localement la section de la poutre.
- ρ_{zone}: la densité adaptée dans les zones pénalisées pour préserver la distribution de masse, notamment en dynamique.

L'ensemble de la méthodologie est évalué sur des cas statiques, dynamiques, non linéaires et expérimentaux. Le chapitre démontre l'efficacité de cette approche à travers des simulations de cellules unitaires et de structures complètes.

7.3.2 Méthodologie

La stratégie développée repose sur l'utilisation d'un modèle poutre de type Timoshenko en 3D (Pilkey 2002), capable de considérer simultanément les effets de flexion et de cisaillement, ce qui est adapté pour les structures lattices avec de faibles rapports d'élancement.

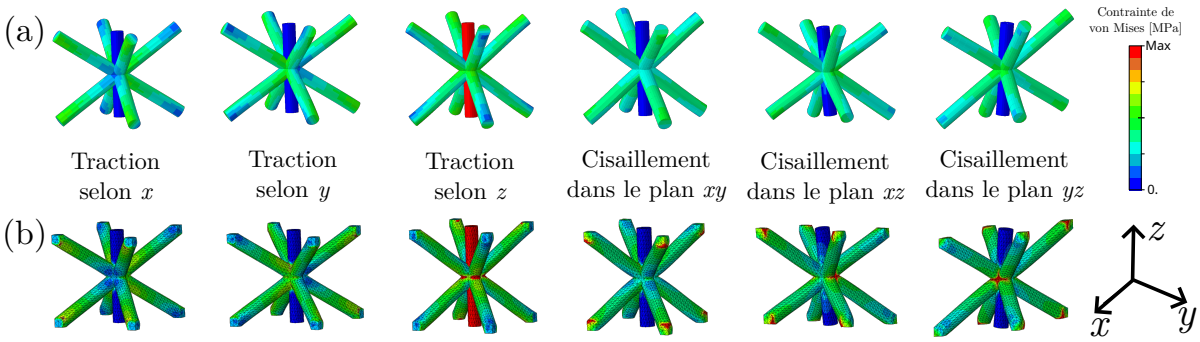

Fig. 7.3 Contrainte de von Mises liée aux six cas de chargement pour la cellule BCCZ : (a) modèle poutre et (b) modèle solide 3D.

Modèle de référence

Afin d'évaluer la précision des modèles poutres, leurs résultats sont systématiquement comparés à ceux de modèles volumiques 3D, considérés comme référence. Les simulations sont menées sur des cellules unitaires en appliquant des conditions aux limites périodiques PBCs, afin d'assurer la représentativité du comportement à l'échelle macroscopique. Six cas de chargement sont analysés : trois tractions uniaxiales dans les directions principales, ainsi que trois cisaillements dans les plans XY, XZ et YZ (Voir Figure 7.3). Les tenseurs d'élasticité effectifs sont calculés à partir des résultats obtenus via des simulations éléments finis réalisées avec le logiciel Abaqus (Dassault Systèmes, Vélizy-Villacoublay, France, Version 2022). Cinq géométries de cellules unitaires, choisies pour leur diversité topologique, ont été sélectionnées afin de couvrir un large spectre de configurations structurelles. Elles sont étudiées pour des densités relatives comprises entre 0.03 et 0.45, afin de déterminer l'impact de la densité sur la précision des modèles poutres. Le matériau utilisé est VeroClear™, un polymère utilisé dans les procédés de fabrication additive PolyJet de Stratasys® (Stratasys, Eden Prairie, MN, USA) (*VeroClear* 2024).

Stratégie de correction

Pour améliorer la rigidité des nœuds, l'approche consiste à pénaliser localement les extrémités des poutres selon deux paramètres :

- Longueur pénalisée l_{zone} : Une zone autour de chaque jonction est définie, dont la longueur dépend du rayon de la poutre R et de l'angle γ entre poutres connectées (Voir Figure 7.4):

$$l_{\text{zone}}(R, \gamma) = \frac{R}{\tan(\gamma/2)}. \tag{7.1}$$

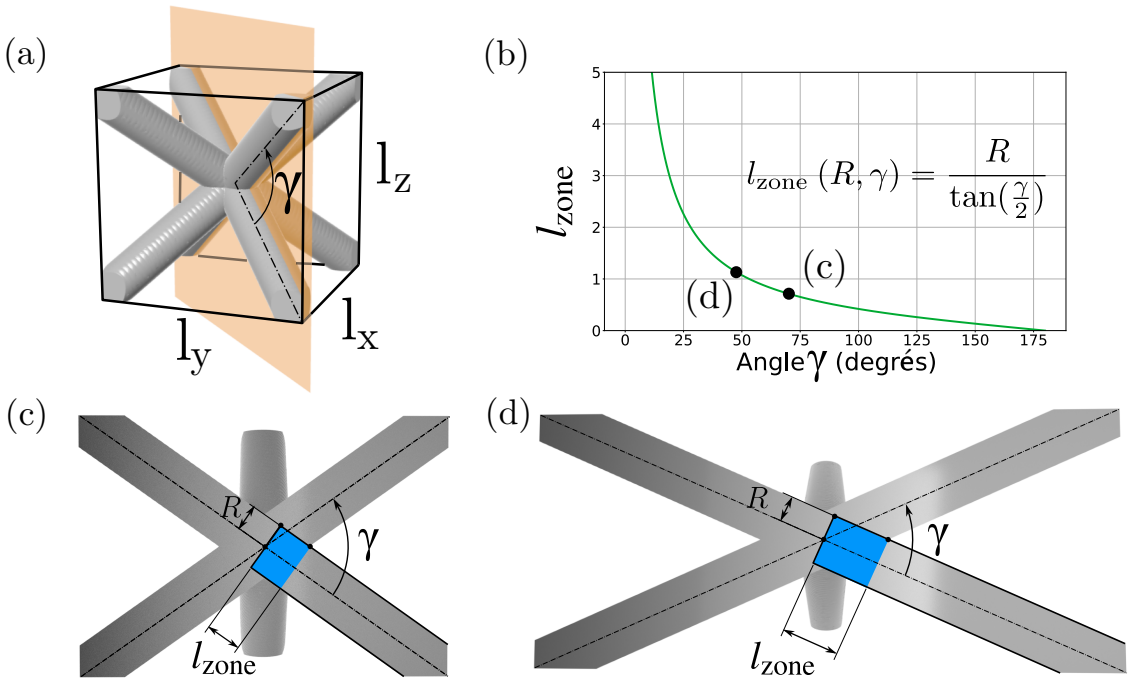

Fig. 7.4 (a) Cellule BCC coupée par un plan central. (b) Graphique illustrant la relation entre l_{zone} et l'angle γ avec $R = 0.5$ mm (Stratégie 4). (c) Vue en coupe de la cellule unitaire BCC ($l_x = l_y = l_z = 5$ mm, $R = 0.5$ mm ce qui implique $\gamma = 70.5°$). (d) Vue en coupe de la cellule unitaire BCC ($l_x = l_y = 8$ mm $l_z = 5$ mm, $R = 0.5$ mm ce qui implique $\gamma = 47.7°$).

Cette expression tient compte de la géométrie réelle du nœud, contrairement à d'autres méthodes qui fixent l_{zone} pour une géométrie prédéfinie.

- Coefficient de pénalisation du rayon α_{r} : Le rayon est multiplié par α_{r} dans la zone pénalisée, augmentant ainsi la section et donc la rigidité locale. Une optimisation par la méthode de Brent est réalisée pour identifier la valeur de α_{r} qui minimise l'erreur entre le modèle poutre et le modèle solide.

Correction de la masse

L'augmentation du rayon implique une augmentation locale de masse, ce qui modifie la réponse dynamique. Une correction de densité est donc appliquée dans les zones pénalisées, calculée pour préserver la masse totale cible (issue du modèle CAD ou d'une référence préétablie). La densité corrigée ρ_{zone}est obtenue analytiquement en fonction des volumes internes et pénalisés du modèle dans le cas de matériau homogène avec la

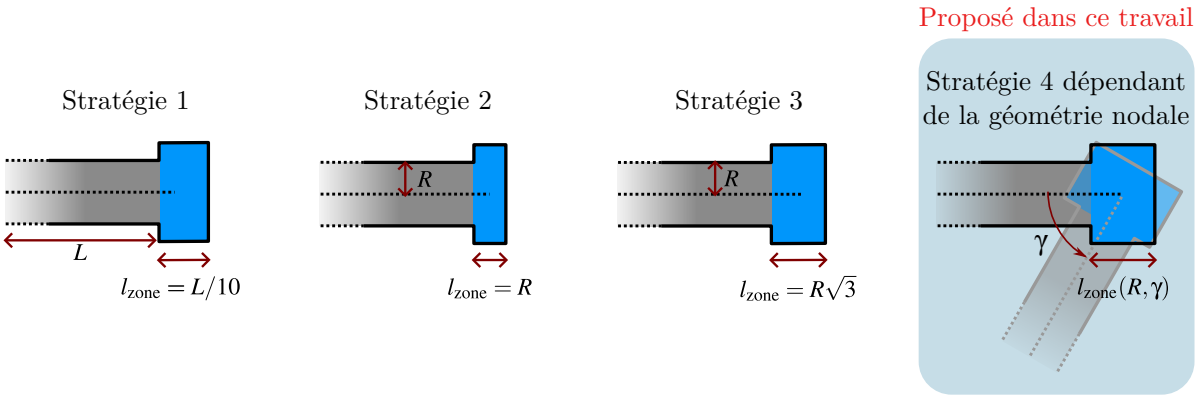

Fig. 7.5 Comparaison schématique des trois stratégies de la littérature et de notre approche pour l_{zone}. (a) Stratégie 1 (Labeas and Sunaric 2010), (b) Stratégie 2 (Gümrük and Mines 2013), (c) Stratégie 3 (H. Guo et al. 2020), (d) Notre approche.

relation :

$$\rho_{\text{zone}} = \frac{1}{V_{\text{zone}}}(m_{\text{obj}} - \rho V_{\text{int}}), \tag{7.2}$$

avec V_{zone} et V_{int} sont les volumes des régions pénalisées et internes du modèle poutre, respectivement, et m_{obj} est la masse totale cible.

Extension non linéaire et expérimentation

Afin d'étendre la compréhension des limites de la méthode de pénalisation, des simulations avec un matériau élasto-plastique et des grandes déformations sont réalisées. De plus, une validation expérimentale est effectuée sur des structures lattices imprimées en 3D via la technologie PolyJet, ce qui permet de vérifier la pertinence de la méthode de pénalisation dans un contexte réel de fabrication additive.

Résultats des simulations statiques

L'indicateur d'erreur principal est une erreur relative globale sur les tenseurs d'élasticité effectifs, calculée via la norme de Frobenius, tel que :

$$\varepsilon_{\text{static}} = \frac{\|\mathbf{S}^{\text{solid}} - \mathbf{S}^{\text{beam}}\|_F}{\|\mathbf{S}^{\text{solid}}\|_F}, \tag{7.3}$$

avec $\|\cdot\|_F$ la norme de Frobenius, et $\mathbf{S} \in \mathbb{R}^{6\times 6}$ représentant la matrice des contraintes effectives vectorisées pour les six cas de chargement. Cette erreur mesure l'écart entre la réponse du modèle poutre (avec ou sans pénalisation) et celle du modèle solide.

Quatre stratégies sont comparées afin de déterminer la plus performante (Voir Figure 7.5) :

- Stratégie 1: $l_{\text{zone}} = L/10$ (L = longueur de la poutre) (Labeas and Sunaric 2010),
- Stratégie 2: $l_{\text{zone}} = R$ (Gümrük and Mines 2013),
- Stratégie 3: $l_{\text{zone}} = R\sqrt{3}$ (H. Guo et al. 2020),
- Stratégie 4: $l_{\text{zone}}(R, \gamma) = \frac{R}{\tan(\gamma/2)}$ (proposée dans ces travaux).

Les résultats montrent que les modèles poutres sans correction produisent des erreurs croissantes avec la densité relative, atteignant jusqu'à 50% pour les densités élevées. La Stratégie 4, considérant la géométrie des jonctions via l'angle γ, est la plus robuste : elle offre les meilleures performances quelle que soit la géométrie.

La détermination du coefficient α_{r} repose sur une approche différente, consistant à utiliser une méthode d'optimisation visant à minimiser l'erreur relative obtenue en simulation statique. L'optimisation est réalisée pour chaque géométrie et chaque densité relative. Les valeurs optimales α_{r}^* se situent entre 1.3 et 1.6 et augmentent avec la densité relative. L'objectif est de réduire l'erreur pour les densités relatives hautes, la moyenne des valeurs optimales pour la densité maximale a été choisie comme valeur générique de référence, soit $\alpha_{\text{r}} = 1.5$.

Résultats des simulations analyse modale

Après validation de la méthodologie en statique, l'étude est étendue à des analyses modales afin de vérifier la capacité des modèles poutres pénalisés à reproduire correctement le comportement dynamique des structures lattices. Des calculs de vibrations naturelles sont réalisés sur les mêmes cellules unitaires, avec conditions périodiques, en comparant les périodes propres obtenues avec :

- le modèle poutre classique (sans pénalisation),
- le modèle poutre pénalisé selon la stratégie 4,
- le modèle poutre pénalisé selon la stratégie 4 et la correction de masse (Equation 7.2).

Une erreur relative dynamique est définie en comparant les vecteurs de périodes naturelles des premiers modes (hors modes rigides). Cet indicateur permet d'évaluer la précision des fréquences basses, essentielles en vibroacoustique ou sous impact.

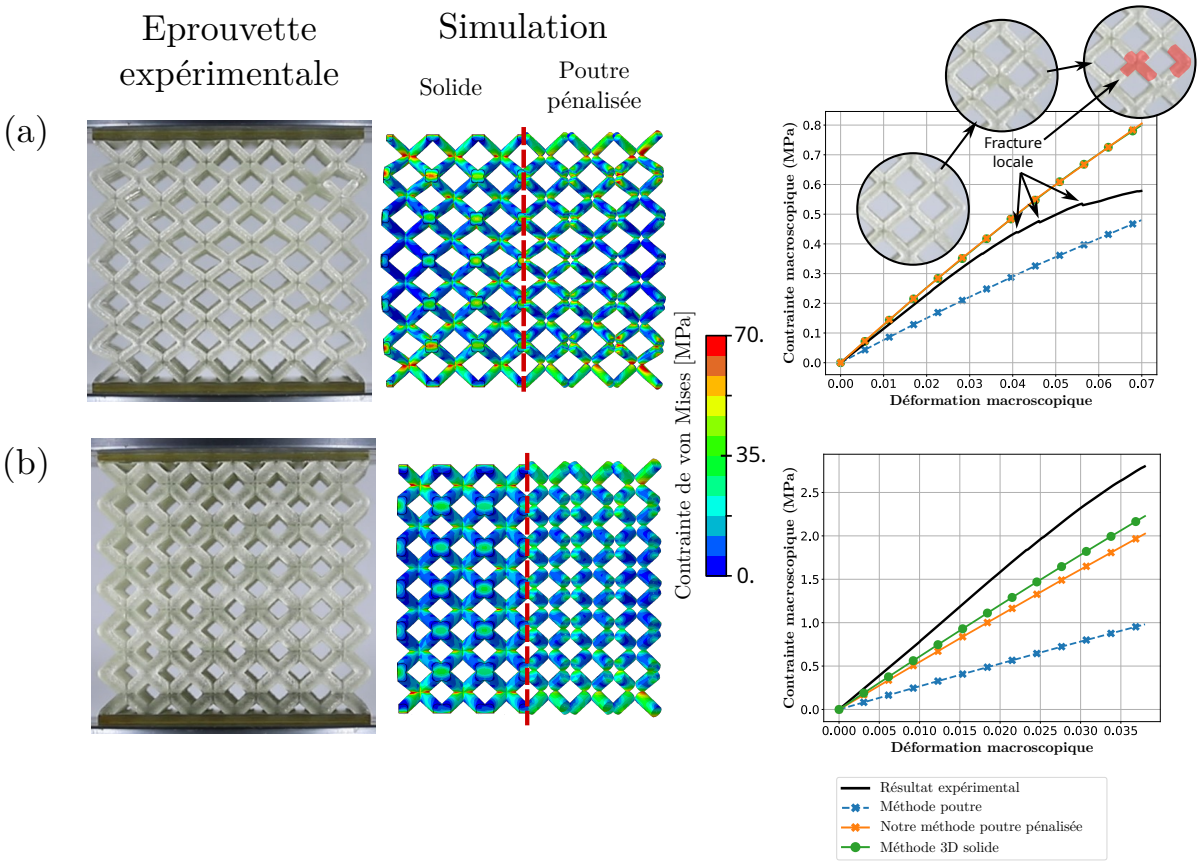

Fig. 7.6 Résumé des résultats sur structure lattice BCC $6 \times 6 \times 6$ (a) $\rho_{\text{rel}} = 0.17$ (b) $\rho_{\text{rel}} = 0.34$.

Les résultats montrent que le modèle poutre classique présente des erreurs importantes, surtout à haute densité relative. Le modèle poutre pénalisé corrige la raideur, mais introduit des erreurs significatives dans les fréquences naturelles. L'élargissement des sections de poutres dans les zones pénalisées induit une augmentation locale de masse, faussant le comportement inertiel global. Le modèle avec pénalisation et correction de masse offre les meilleures performances, avec des erreurs relatives inférieures à 5% pour les densités jusqu'à 0.45.

En conclusion, la pénalisation géométrique améliore la raideur des modèles poutres, mais son effet secondaire sur la masse doit impérativement être corrigé pour des simulations dynamiques fiables.

7.3.3 Extension aux cas non linéaires et validation expérimentale

Après étude de ma méthode en statique et en dynamique linéaire, la robustesse de la méthode est testée dans des contextes plus complexes, intégrant des non-linéarités géométriques, des lois matériaux élasto-plastiques et une comparaison directe avec des essais expérimentaux.

Des simulations de compression sont réalisées sur une cellule BCC modélisée : Les modèles intègrent une loi matériau élasto-plastique, identifiée à partir d'essais de traction sur le matériau VeroClear™, et considèrent les grandes déformations (géométrie non linéaire activée via `NLGEOM` dans Abaqus). Les résultats sont comparés à ceux obtenus avec un modèle solide 3D, considéré comme référence. Les résultats montrent que le modèle poutre classique sous-estime fortement la raideur de la structure, le modèle pénalisé reproduit fidèlement la réponse jusqu'à $\rho_{\text{rel}} \approx 0.3$. Au-delà de cette densité, des écarts apparaissent, dus à l'incapacité du modèle poutre à capturer certains effets plastiques et de localisation de la contrainte.

Des structures lattices complètes BCC ($6 \times 6 \times 6$ cellules, imprimées en PolyJet) ont été fabriquées et testées en compression. Deux configurations ont été évaluées :

- $R = 1.0$ mm $\Rightarrow \rho_{\text{rel}} = 0.17$: dans le domaine de validité du modèle,
- $R = 1.5$ mm $\Rightarrow \rho_{\text{rel}} = 0.34$: hors domaine de validité.

À basse densité, la méthode pénalisée correspond très bien aux essais expérimentaux, surpassant le modèle poutre classique. À forte densité, l'écart entre simulation et expérience augmente, même pour le modèle solide, en raison des limites de la loi matériaux et de la complexité des ruptures locales. Néanmoins, le modèle poutre pénalisé présente un gain de temps de calcul très significatif : plus de 14 heures sont nécessaires pour le modèle solide, contre moins de 2 minutes pour le modèle poutre modifié, et ce sans compromettre significativement la précision des résultats.

7.3.4 Conclusion

Ce chapitre propose une méthodologie générique, non-intrusive et efficace pour améliorer la précision des modèles poutres dans la simulation des structures lattices. Elle repose sur une correction géométrique locale des jonctions, simple à implémenter dans n'importe quel code éléments finis (comme Abaqus), et sans modification du solveur.

Trois paramètres structurent la méthodologie, comme illustré dans la Figure 7.7 :

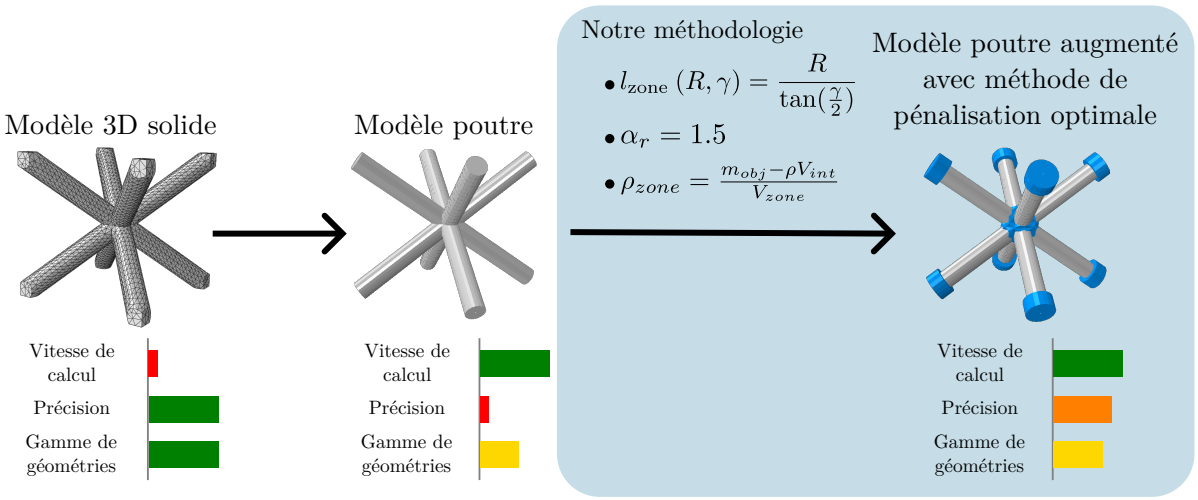

Fig. 7.7 Aperçu de la méthode de pénalisation optimale dans les modèles de poutres

- Longueur de zone pénalisée : $l_{\text{zone}}(R,\gamma) = \frac{R}{\tan(\gamma/2)}$ permet de définir une zone avec rigidité augmentée aux jonctions, adaptée à la géométrie locale.
- Coefficient de pénalisation du rayon : $\alpha_{\text{r}} = 1.5$ donne une rigidification optimale moyenne pour les géométries testées.
- Correction de masse dans les zones pénalisées : $\rho_{\text{zone}} = \frac{1}{V_{\text{zone}}}(m_{\text{obj}} - \rho V_{\text{int}})$ permet de conserver une distribution de masse proche du cas réel pour la simulation en dynamique.

Cette approche est testée sur des cellules unitaires variées, en simulations statiques, dynamiques, non linéaires, et validée expérimentalement. La méthodologie est particulièrement efficace pour des densités relatives inférieures à 0.3, qui couvrent la majorité des applications lattices. Au-delà, la précision baisse, mais reste bien supérieure à celle des modèles poutres classiques. En résumé, cette méthode fournit une solution simple, robuste et généralisable pour corriger les limites connues des modèles poutres, tout en conservant leur légèreté numérique.

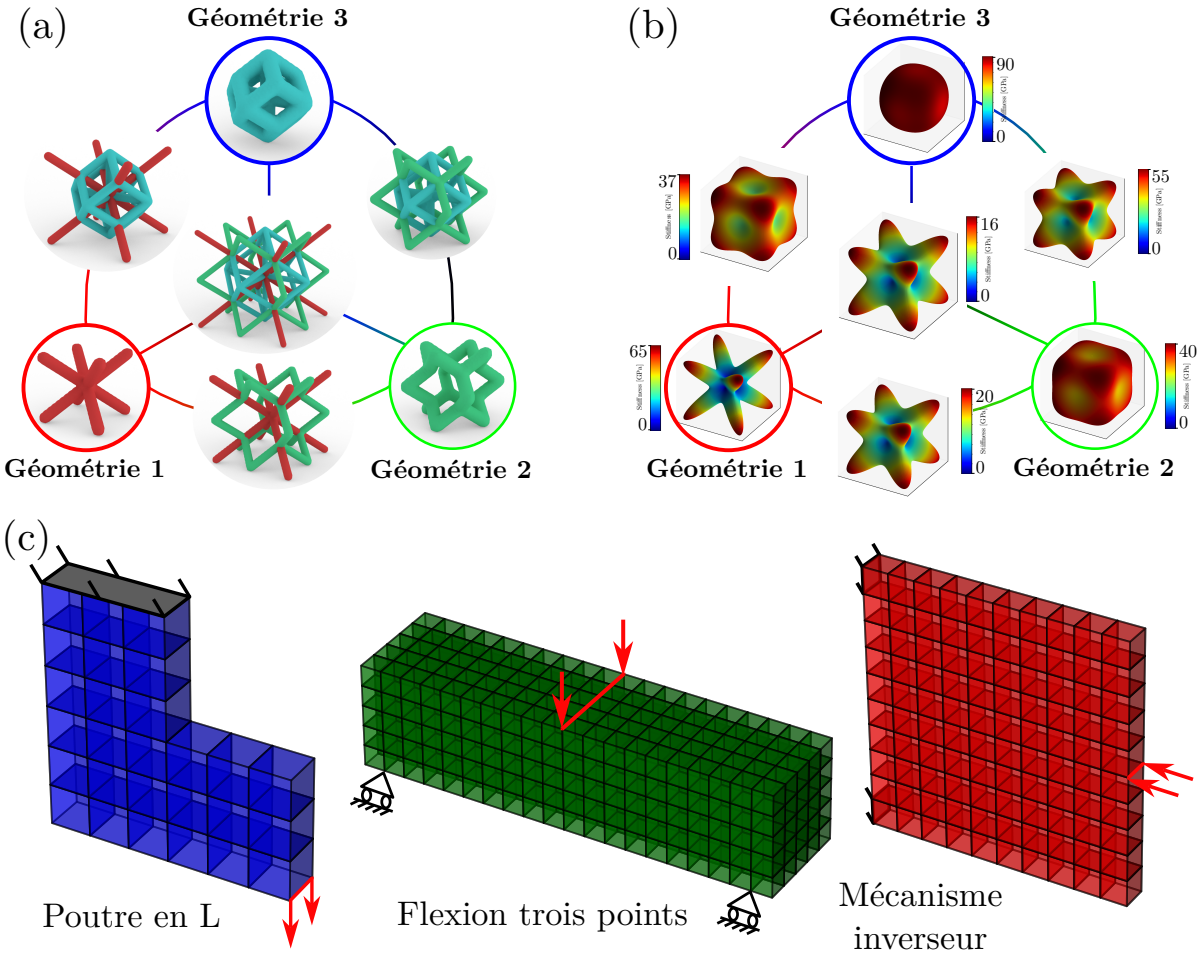

Fig. 7.8 Illustration des éléments constitutifs et des cas d'étude utilisés pour l'évaluation de la méthode proposée. (a) Géométries de base sélectionnées pour la construction des cellules hybrides, combinant trois configurations primitives distinctes. (b) Comportements mécaniques homogénéisés associés à chaque géométrie de base, mettant en évidence la diversité des réponses en termes de rigidité équivalente. (c) Trois configurations structurelles de référence utilisées comme référence : poutre en L encastrée, essai de flexion trois points et mécanisme inverseur.

7.4 Approche de décomposition de domaine assistée par modèle de substitution pour la simulation efficace de structures lattices

7.4.1 Fondations pour l'exploration des structures lattices localement optimisées

En premier lieu, le cadre numérique développé pour la conception et la simulation efficace de structures lattices localement optimisées est présenté. L'objectif est de proposer une approche flexible capable d'intégrer divers types de cellules unitaires, paramètres d'optimisation et configurations structurelles. Une méthode de paramétrisation est établie au niveau cellulaire afin de définir un espace de conception à la fois riche et maîtrisable. Les structures hybrides sont retenues car elles combinent plusieurs géométries primitives, permettant d'élargir les comportements mécaniques accessibles tout en limitant le nombre de variables d'optimisation. Trois géométries de base ont été sélectionnées selon des critères de compatibilité, d'indépendance, d'hétérogénéité et d'imprimabilité (Voir Figure 7.8(a)). Cette approche assure la continuité géométrique entre cellules et favorise une exploration efficace de l'espace de conception. Trois configurations de référence sont définies pour évaluer la précision et la robustesse des modèles, comme présenté dans la Figure 7.8(c). Ces cas couvrent des situations de complexité croissante et serviront de base à la validation des modèles de substitution présentés dans la suite du chapitre.

7.4.2 Méthode de décomposition de domaine appliquée aux structures lattices

Formulation de la méthode

Dans ce travail, la méthode de décomposition de domaine équilibrée est employée. La Figure 7.9 illustre les différentes étapes de la Méthode de Décomposition de Domaine (MDD) appliquée à une structure lattice bidimensionnelle composée de 2×2 cellules unitaires. Le processus itératif peut être décrit comme suit :

1. **Définition du problème** : définition du problème global, incluant la géométrie, les conditions aux limites et les conditions de chargement.
2. **Décomposition du domaine** : subdivision du problème global en sous-domaines plus petits, correspondant ici aux cellules de la structure lattice.

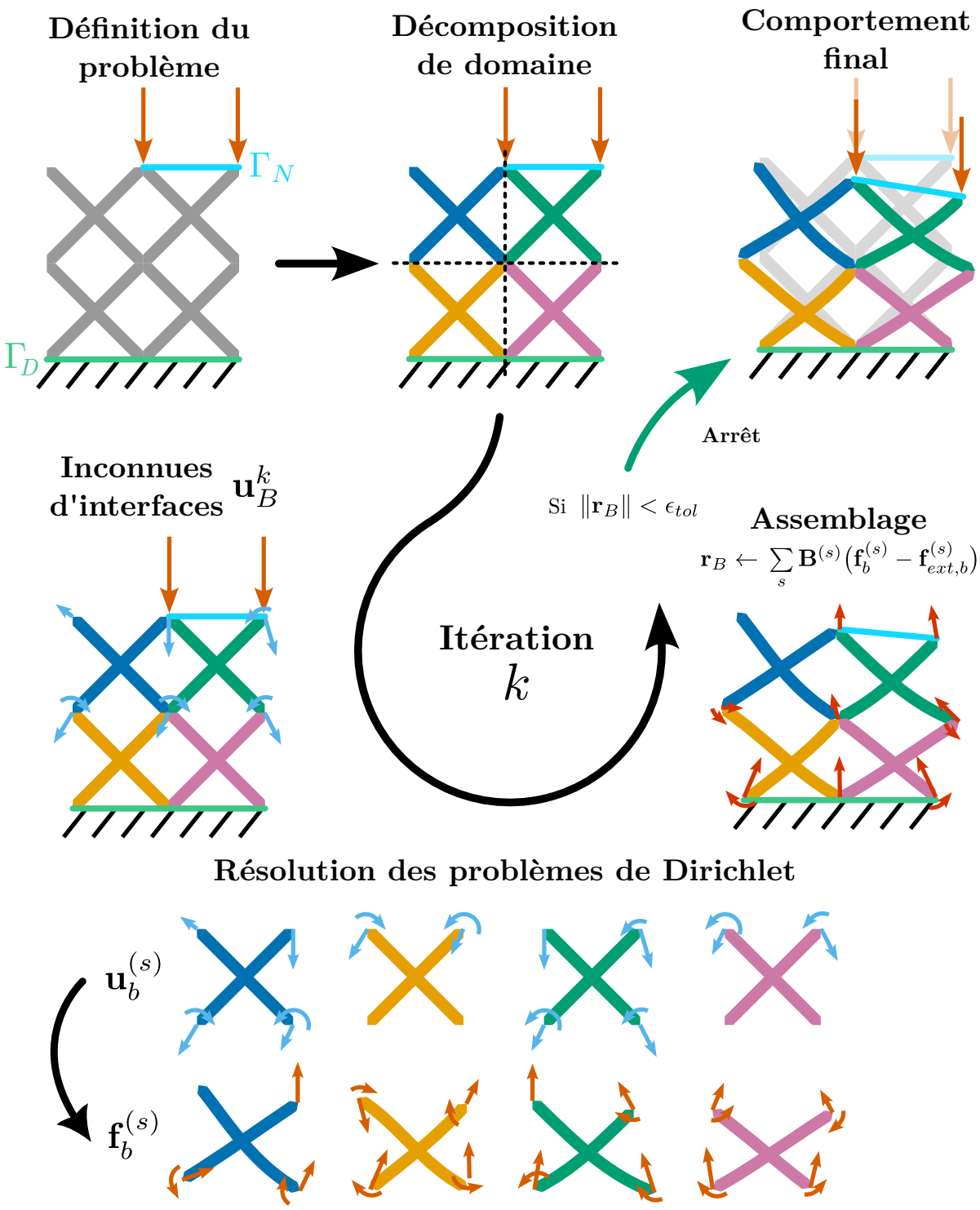

Fig. 7.9 Représentation schématique, étape par étape, de la méthode de décomposition de domaine équilibrée appliquée à des structures lattices BCC bidimensionnelles. Chaque nœud possède trois degrés de libertés, correspondant à deux translations et une rotation.

3. **Initialisation des inconnues d'interface** : attribution d'une valeur initiale pour les déplacements et rotations associés aux nœuds situés sur les interfaces

4. **Résolution indépendante de chaque sous-domaine** : calcul de l'équilibre de chaque sous-domaine en résolvant un problème de Dirichlet

5. **Assemblage des forces et moments de réaction aux interfaces** : calcul des efforts de réaction locaux et vérification de l'équilibre des charges aux interfaces.

6. **Mise à jour des variables d'interface** : si le résidu des efforts d'interface est inférieur à un seuil de tolérance, la solution globale est atteinte ; sinon, la procédure est répétée en actualisant les inconnues d'interface à partir du résidu

Implémentation de la méthode

La MDD présente plusieurs avantages majeurs qui la rendent particulièrement adaptée à l'analyse des structures lattices :

- **Réduction de la taille du problème** : En décomposant le problème global en sous-problèmes locaux plus petits, la méthode réduit la dimension globale du système, rendant les calculs plus légers et moins exigeants en mémoire. Le calcul du complément de Schur au niveau de chaque sous-domaine condense davantage le système aux seuls degrés de liberté d'interface, nettement moins nombreux que l'ensemble des degrés de liberté du modèle complet.

- **Flexibilité géométrique et fonctionnelle** : La méthode peut s'adapter à une large variété de géométries lattices et de conditions de chargement, ce qui renforce sa capacité d'application à différents problèmes de conception et d'optimisation.

- **Compatibilité avec des physiques hétérogènes** : Chaque sous-domaine pouvant être associé à ses propres propriétés matériaux et conditions physiques, la méthode permet de traiter des géométries complexes et des comportements multi-physiques dans un cadre cohérent et modulaire.

- **Avantages spécifiques aux structures poutres** : La méthode est particulièrement adaptée aux structures de type poutre, car les degrés de liberté d'interface sont naturellement définis aux nœuds de connexion des poutres, ce qui simplifie la formulation et l'implémentation. Ce choix évite également les problèmes de maillage non conforme rencontrés avec des géométries cellulaires plus complexes.

- **Parallélisation naturelle** : Chaque sous-domaine pouvant être résolu indépendamment, la méthode se prête aisément au calcul parallèle, ce qui permet de réduire significativement le temps global de simulation.

Table 7.1 Expériences numériques illustrant la précision et l'efficacité de calcul des différentes approches de simulation.

Méthode	**Type de structure**	**Précondi-tionneur**	**Temps de calcul (s)**	**Erreur relative**	**Nombre d'itérations**
Modèle global éléments finis	Périodique	—	506.2	Référence	—
Modèle global éléments finis	Aléatoire	—	512.3	Référence	—
MDD standard	Périodique	✗	58.8	$\approx 10^{-6}$	465
MDD standard	Périodique	✓	14.5	$\approx 10^{-11}$	1
MDD standard	Aléatoire	✗	665.4	$\approx 10^{-6}$	5702
MDD standard	Aléatoire	✓	47.3	$\approx 10^{-6}$	275
MDD standard pénalisation erronée	Aléatoire	✓	59.5	$\approx 10^{-3}$	209
MDD périodique	Périodique	✗	49.2	$\approx 10^{-6}$	466
MDD périodique	Périodique	✓	5.2	$\approx 10^{-11}$	1
MDD périodique	Aléatoire	—	—	—	—

L'implémentation de la méthode de décomposition de domaine a été validée à partir du cas de référence de la flexion trois points, en comparaison avec les simulations éléments finis globales. Trois approches ont été évaluées : la simulation à l'échelle globale, la MDD standard, et la version périodique exploitant la répétitivité des cellules.

Les résultats présentés dans le Tableau 7.1 montrent que la méthode standard permet une réduction de près de 97 % du temps de calcul par rapport au modèle complet, tout en conservant une précision de l'ordre de 10^{-6}. L'utilisation d'un préconditionneur

basé sur les compléments de Schur locaux exacts permet une convergence en une seule itération, puisque le problème d'interface est alors directement résolu. Pour les structures non périodiques, une légère déviation vis-à-vis du modèle complet est observée. Cette différence provient principalement du traitement des poutres en frontière, dont la pénalisation est calculée indépendamment dans chaque cellule, sans prise en compte de l'influence des voisines. L'introduction d'un mappage périodique au niveau cellulaire et structurel améliore considérablement cette cohérence mécanique. Dans le cas périodique, le calcul d'un unique complément de Schur suffit, ce qui simplifie encore la phase de prétraitement et accélère les calculs d'un facteur proche de deux ordres de grandeur. Sans préconditionneur, la convergence reste correcte mais plus lente, confirmant l'efficacité du préconditionnement exact.

De manière générale, la méthode de décomposition de domaine démontre une excellente efficacité numérique pour la simulation de structures lattices, à condition que la formulation et le calcul des compléments de Schur soient correctement définis. La principale limite reste le coût de calcul associé à ces compléments pour chaque géométrie de cellule. Pour y remédier, l'utilisation de modèles réduits ou de modèles de substitution constitue une perspective prometteuse, permettant d'estimer les compléments de Schur de manière continue et différentiable sur l'ensemble de l'espace des paramètres.

7.4.3 Modèle de substitution pour l'analyse rapide

Cette section introduit un modèle de substitution visant à accélérer la simulation numérique en approchant les compléments de Schur locaux dépendant des paramètres géométriques. La méthode repose sur la réduction de modèle par bases réduites, enrichie par des techniques d'hyper-réduction et un algorithme glouton (greedy) permettant de sélectionner de manière itérative les configurations les plus représentatives. Cette approche assure la construction d'une base compacte et pertinente, garantissant un compromis optimal entre vitesse de calcul et précision de l'approximation.

Divers modèles de substitution

Afin de disposer d'un modèle de substitution à la fois rapide et précis, plusieurs stratégies d'approximation ont été mises en œuvre pour évaluer efficacement les coefficients réduits $\alpha_i(\boldsymbol{\mu})$ dépendant des paramètres géométriques. Ces approches diffèrent par leur complexité, leur coût de calcul et leur capacité à restituer les variations locales de la fonction à approcher, ce qui conditionne directement leur pertinence selon le niveau de

précision et le domaine d'application visés. Elles vont des méthodes les plus simples, comme l'interpolation du plus proche voisin ou l'interpolation barycentrique linéaire, jusqu'à des approches plus avancées telles que les fonctions à base radiale ou les réseaux de neurones artificiels.

Les deux premières méthodes présentent l'avantage d'une mise en œuvre directe et d'un coût de calcul très faible, mais leur précision dépend fortement de la densité des points d'interpolation. L'interpolation par fonctions à base radiale permet de mieux capturer les variations continues au sein de l'espace des paramètres tout en assurant une représentation lisse, au prix d'un coût numérique plus élevé. Enfin, les réseaux de neurones offrent une alternative puissante pour modéliser des relations fortement non linéaires, en apprenant directement la dépendance entre les paramètres géométriques et les coefficients réduits, mais nécessitent une phase d'entraînement plus coûteuse et un jeu de données représentatif.

Ces différentes techniques sont comparées dans la suite en termes de précision, de robustesse et de coût de calcul, afin d'identifier la stratégie la plus adaptée à l'intégration dans le cadre de la méthode de décomposition de domaine.

Évaluation des performances des modèles de substitution

L'évaluation des modèles de substitution vise à analyser leur capacité à reproduire fidèlement la variation des coefficients réduits $\alpha_i(\boldsymbol{\mu})$ et à prédire le comportement global des structures sur différents cas tests.

Analyse des coefficients réduits La Figure 7.10(a) illustre la variation des coefficients réduits pour une géométrie donnée, obtenue avec la tolérance gloutonne $\varepsilon_{\text{greedy}} = 10^{-3}$. Les coefficients présentent globalement une évolution régulière en fonction du paramètre géométrique. Les méthodes d'interpolation les plus simples, comme le plus proche voisin ou l'interpolation linéaire, produisent respectivement des approximations discontinues ou seulement continues en C^0, ce qui limite leur utilisation pour des stratégies d'optimisation basées sur le gradient. À l'inverse, les modèles à base de fonctions radiales et de réseaux de neurones assurent une représentation plus régulière, respectivement C^1 et C^∞, garantissant la différentiabilité nécessaire à la propagation des sensibilités dans le processus d'optimisation.

Validation sur cas tests Les performances globales ont été évaluées sur plusieurs cas de référence, dont ceux introduits à la Section 4.1. La Figure 7.10(a) présente

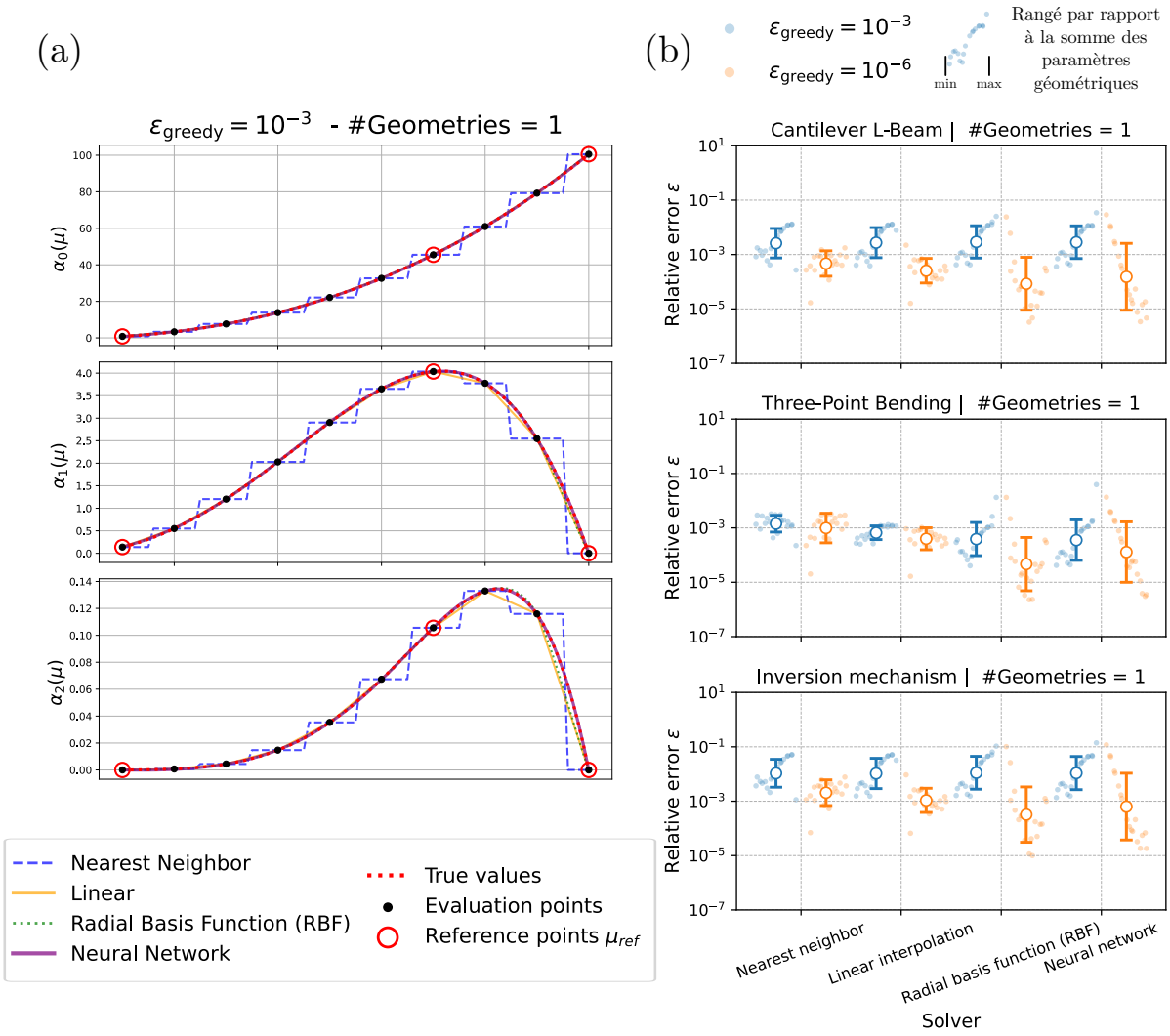

Fig. 7.10 Comparaison des performances des modèles de substitution. (a) Évolution des coefficients réduits $\alpha_i(\boldsymbol{\mu})$, (b) Erreurs relatives globales obtenues sur les cas tests de référence (poutre en L en porte-à-faux, flexion trois points et mécanisme d'inversion) pour deux tolérances gloutonnes $\varepsilon_{\text{greedy}} = 10^{-3}$ et $\varepsilon_{\text{greedy}} = 10^{-6}$.

la distribution des erreurs relatives sur les déplacements et rotations aux frontières, comparées aux simulations éléments finis de référence. Les résultats montrent que les modèles à base de fonctions radiales et de réseaux de neurones offrent la meilleure précision, avec des erreurs moyennes inférieures à 10^{-3}. Les méthodes les plus simples restent plus stables mais moins précises, tandis que les modèles avancés reproduisent plus finement les variations locales, au prix d'une variabilité légèrement accrue selon les configurations. De plus, la précision dépend du type de problème étudié : les erreurs sont en moyenne plus faibles pour le cas de flexion trois points que pour le mécanisme

d'inversion, en raison du nombre plus élevé de cellules impliquées qui lissent la réponse globale.

Bilan général Dans l'ensemble, les résultats confirment que les modèles à base de fonctions radiales et de réseaux de neurones constituent les approches les plus performantes pour intégrer la substitution dans le cadre de la décomposition de domaine. Les premiers assurent un compromis favorable entre coût et précision, tandis que les seconds garantissent la régularité et la différentiabilité nécessaires aux approches d'optimisation basées sur le gradient. Ainsi, ces modèles s'avèrent particulièrement adaptés pour la conception optimale de structures lattices à géométries indépendantes.

7.4.4 Modèle de substitution augmenté par la physique

Dans le cas linéaire, petites déformations et quasi-statique, le complément de Schur ne dépend que des paramètres géométriques $\boldsymbol{\mu}$. Dès que des effets dynamiques ou harmoniques interviennent, des dépendances additionnelles au temps ou à la fréquence apparaissent, ce qui augmente fortement la dimension de l'espace des paramètres et rend les méthodes de réduction par projection moins efficaces. Afin d'éviter cette augmentation dimensionnelle, nous adoptons un modèle de substitution non linéaire de type réseau de neurone augmenté par la physique, intégrant directement des connaissances physiques pour réduire les besoins en données et améliorer la généralisation du modèle. L'objectif est d'approximer la loi d'interface reliant déplacements imposés et efforts d'interface au niveau d'une cellule. Plutôt que d'apprendre directement la carte non linéaire $\mathbf{f}(\mathbf{u}_b; \boldsymbol{\mu})$, le réseau prédit l'énergie de déformation $\mathcal{E}(\mathbf{u}_b; \boldsymbol{\mu})$, les charges d'interface étant ensuite obtenues par différentiation $\mathbf{f} = \partial\mathcal{E}/\partial\mathbf{u}_b$ grâce à la différentiation automatique. Cette formulation basée sur l'énergie permet de réduire la dimension de sortie à une variable scalaire, tout en préservant les propriétés fondamentales du système, telles que la symétrie, l'équilibre et l'annulation des efforts pour les mouvements de corps rigide.

Architecture du réseau. L'architecture PANN est conçue de manière à exploiter explicitement la structure physique du problème. L'architecture est illustré dans la Figure 7.11 En entrée, le réseau reçoit deux ensembles de variables : (i) les paramètres géométriques $\boldsymbol{\mu}$, et (ii) un vecteur d'*énergies de référence* $\boldsymbol{\mathcal{E}}_{\text{ref}}(\mathbf{u}_b)$ calculé à partir de compléments de Schur pré-sélectionnés par un algorithme glouton, jouant le rôle d'un espace latent physiquement informé. Ces deux blocs d'entrée sont concaténés et traités par plusieurs couches denses à activations lisses (par exemple `tanh`), afin d'assurer la

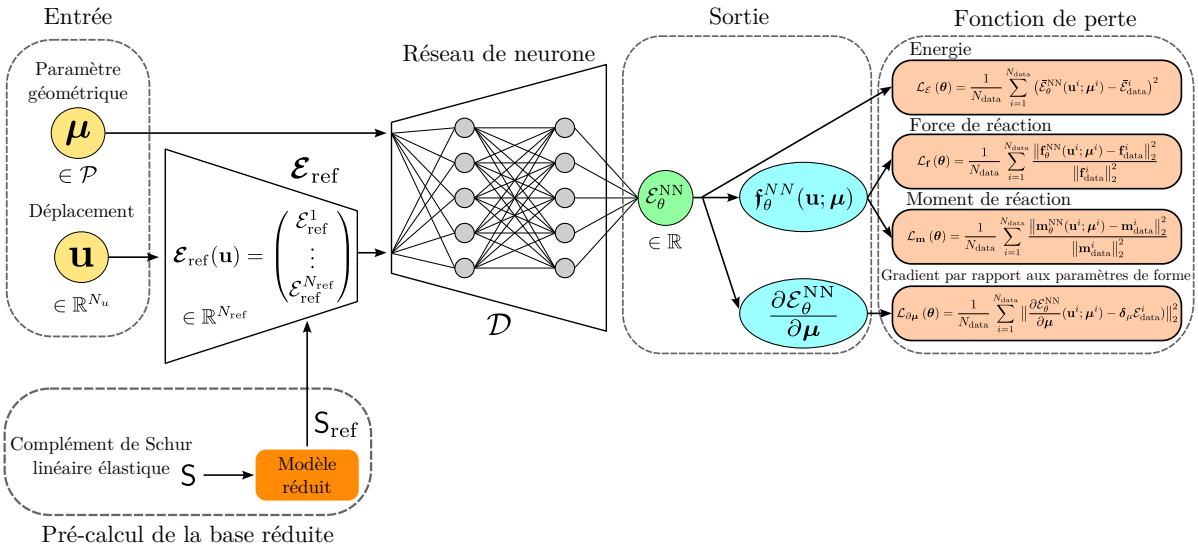

Fig. 7.11 Architecture du modèle PANN développé.

continuité des dérivées nécessaires à l'optimisation. La sortie du réseau correspond à un scalaire représentant l'énergie interne prédite $\mathcal{E}^{\text{NN}}_\theta(\mathbf{u}_b;\boldsymbol{\mu})$. Les efforts et moments d'interface utilisés par la méthode de décomposition sont ensuite obtenus par :

$$\mathfrak{f}^{\text{NN}} = \frac{\partial \mathcal{E}^{\text{NN}}_\theta}{\partial \mathbf{u}_b} = \frac{\partial \mathcal{E}^{\text{NN}}_\theta}{\partial \boldsymbol{\mathcal{E}}_{\text{ref}}} \cdot \frac{\partial \boldsymbol{\mathcal{E}}_{\text{ref}}}{\partial \mathbf{u}_b}, \quad \text{avec} \quad \frac{\mathrm{d}\mathcal{E}^i_{\text{ref}}}{\mathrm{d}\mathbf{u}_b} = \mathbf{S}(\boldsymbol{\mu}^i_{\text{ref}})\,\mathbf{u}_b. \tag{7.4}$$

Ce schéma garantit la cohérence énergétique du modèle, facilite le calcul des sensibilités pour les approches d'optimisation, et réduit considérablement le nombre de paramètres à estimer.

Résultats L'intégration du modèle PANN dans la méthode de décomposition de domaine conduit initialement à des erreurs importantes, car de nombreux points évalués lors de la résolution se situent en dehors du domaine d'apprentissage du réseau. Cette faible capacité d'extrapolation est inhérente aux réseaux de neurones, qui ne peuvent prédire de manière fiable que dans la plage de données utilisée pour l'entraînement. L'analyse montre que les évaluations situées dans le domaine d'apprentissage correspondent principalement aux cellules proches des conditions aux limites, tandis que celles issues des itérations du gradient conjugué s'en écartent progressivement. L'adoption d'un solveur par gradient conjugué non linéaire (NCG) permet de réaligner les évaluations sur des états physiquement cohérents, réduisant ainsi les erreurs d'environ un ordre de grandeur. Enfin, une stratégie de linéarisation locale du problème améliore encore les résultats,

avec des erreurs désormais inférieures à 1%, voire proches de 0.1% pour le cas de flexion trois points.

7.4.5 Discussion et conclusion

Ce chapitre a présenté une approche hybride combinant la méthode de décomposition de domaine et des modèles de substitution pour la simulation efficace de structures lattices à géométrie localement variable. Deux familles de modèles ont été développées : les modèles basés sur la réduction de base et l'interpolation des coefficients réduits, et les modèles de réseau de neurone augmentés par la physique, qui apprennent directement l'énergie interne des cellules à partir de principes physiques. Les modèles de réduction de base se sont révélés particulièrement robustes et précis pour les cas linéaires, tandis que les réseaux de neurone offrent une généralisation plus large, notamment pour des comportements non linéaires, au prix d'une plus grande complexité d'apprentissage. Dans l'ensemble, cette méthodologie ouvre la voie à une simulation plus rapide et plus flexible des structures lattices, tout en constituant une base solide pour leur intégration future dans un cadre d'optimisation multi-paramètres.

7.5 Vers une optimisation locale efficace des structures lattices

L'optimisation des structures lattices est particulièrement complexe en raison du fort couplage entre géométrie et performance, qui conduit à des espaces de conception de très grande dimension. Les approches classiques basées sur des structures à gradient, où quelques paramètres globaux (comme le rayon des poutres ou la taille des cellules) contrôlent l'ensemble de la structure, offrent une simplification utile mais restent limitées pour des géométries complexes ou soumises à des chargements multiples. Pour dépasser ces limites, la stratégie proposée permet à chaque cellule de varier indépendamment, définissant ainsi un espace de conception riche mais hautement dimensionnel. Le cadre d'optimisation développé, fondé sur la méthode de décomposition de domaine et les modèles de substitution présentés à la section précédente, permet l'emploi de méthodes d'optimisation à gradient dans cet espace avec des géométries de cellules indépendantes. L'objectif est de maximiser la rigidité globale (ou minimiser la compliance) pour une masse donnée, sous contraintes de densité relative et de bornes géométriques sur les rayons des poutres.

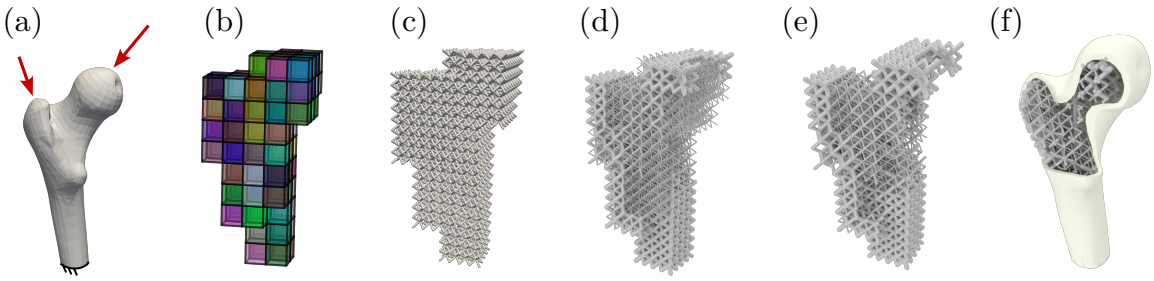

Fig. 7.12 Vue d'ensemble de la méthode proposée appliquée à une prothèse de tête fémorale, comprenant : (a) l'application des conditions aux limites, (b) le pavage de la structure en cellules, (c) le remplissage des cellules avec des géométries lattices, (d) l'optimisation de la structure, (e) le post-traitement du design, et (f) l'adaptation finale de la lattice à la géométrie initiale.

7.5.1 Étude de cas : Chaîne complète d'optimisation appliquée à une prothèse de tête fémorale

Dans les applications réelles, l'espace de conception peut devenir extrêmement complexe, nécessitant des étapes de traitement supplémentaires. La Figure 7.12 illustre les différentes étapes de la méthode appliquée à une prothèse de tête fémorale (fichier STL issu de Leiden (2015)). Cet exemple a été choisi en raison de la complexité géométrique du modèle, de la présence de multiples conditions de chargement favorisant l'utilisation de géométries hybrides, et de son adéquation avec la fabrication additive, la prothèse devant être adaptée à l'anatomie du patient (Naceur et al. 2015).

La méthode proposée pour l'optimisation de structures lattices multifonctionnelles se compose de six étapes principales :

1. **Application des conditions aux limites (a) :** Définir les conditions aux limites sur le maillage global de la structure.

2. **Pavage de la structure (b) :** Discrétiser la géométrie en cellules de taille et de forme adaptées. Plusieurs stratégies peuvent être envisagées pour paver des formes complexes. Dans ce travail, un pavage voxelisé standard est employé, la grille de voxels étant légèrement étendue au-delà des frontières de l'objet afin d'assurer une couverture complète. Des approches plus avancées, à taille ou forme de cellule variable, peuvent également être utilisées, à condition qu'un modèle de substitution soit disponible pour prédire le comportement mécanique de chaque cellule.

3. **Remplissage des cellules (c) :** Chaque cellule est remplie d'une géométrie lattice, qu'elle soit basée sur des poutres, des surfaces ou des volumes. La seule contrainte est de disposer d'un modèle de substitution capable de prédire la réponse mécanique de la géométrie choisie.

4. **Optimisation de la structure (d) :** Réaliser la procédure d'optimisation.

5. **Post-traitement du design (e) :** Appliquer les opérations de post-traitement nécessaires pour garantir la fabricabilité et la cohérence structurelle.

6. **Adaptation finale de la lattice (f) :** Ajuster la lattice à la géométrie initiale complexe. L'approche la plus simple consiste à tronquer la lattice à l'aide du maillage original, mais des techniques plus avancées de morphing ou de déformation peuvent également être envisagées. Idéalement, le morphing devrait être effectué avant l'optimisation afin de préserver la cohérence physique. Dans cette étude, le maillage initial est épaissi pour créer une peau autour de la structure, bien que cette étape ne soit pas prise en compte dans les phases précédentes de la méthode.

Cette chaîne d'optimisation fournit ainsi un cadre complet pour la conception et l'optimisation de structures lattices multifonctionnelles au sein de géométries complexes. En combinant la décomposition de domaine et la modélisation de substitution, la méthode permet de traiter efficacement un grand nombre de variables de conception, la rendant adaptée à des applications industrielles réalistes.

7.5.2 Validation expérimentale sur les structures de référence

La Figure 7.13 illustre l'ensemble du processus expérimental mené pour valider le cadre d'optimisation proposé sur les trois cas tests de référence : la poutre en L en porte-à-faux, la poutre en flexion trois points et le mécanisme inverseur. Pour chacun d'eux, une version de référence à distribution uniforme et une version optimisée obtenue par la méthode développée ont été comparées. La distribution spatiale des géométries de base au sein des structures optimisées met en évidence la capacité du cadre d'optimisation à adapter localement les caractéristiques de chaque cellule selon les sollicitations mécaniques spécifiques : par exemple, dans la poutre en L, la géométrie [2] se concentre à proximité de l'encastrement, tandis que dans la poutre en flexion trois points, les géométries [1] et [3] dominent la zone centrale soumise à la flexion. Ces résultats soulignent le rôle crucial du choix des géométries élémentaires dans la performance finale de la structure.

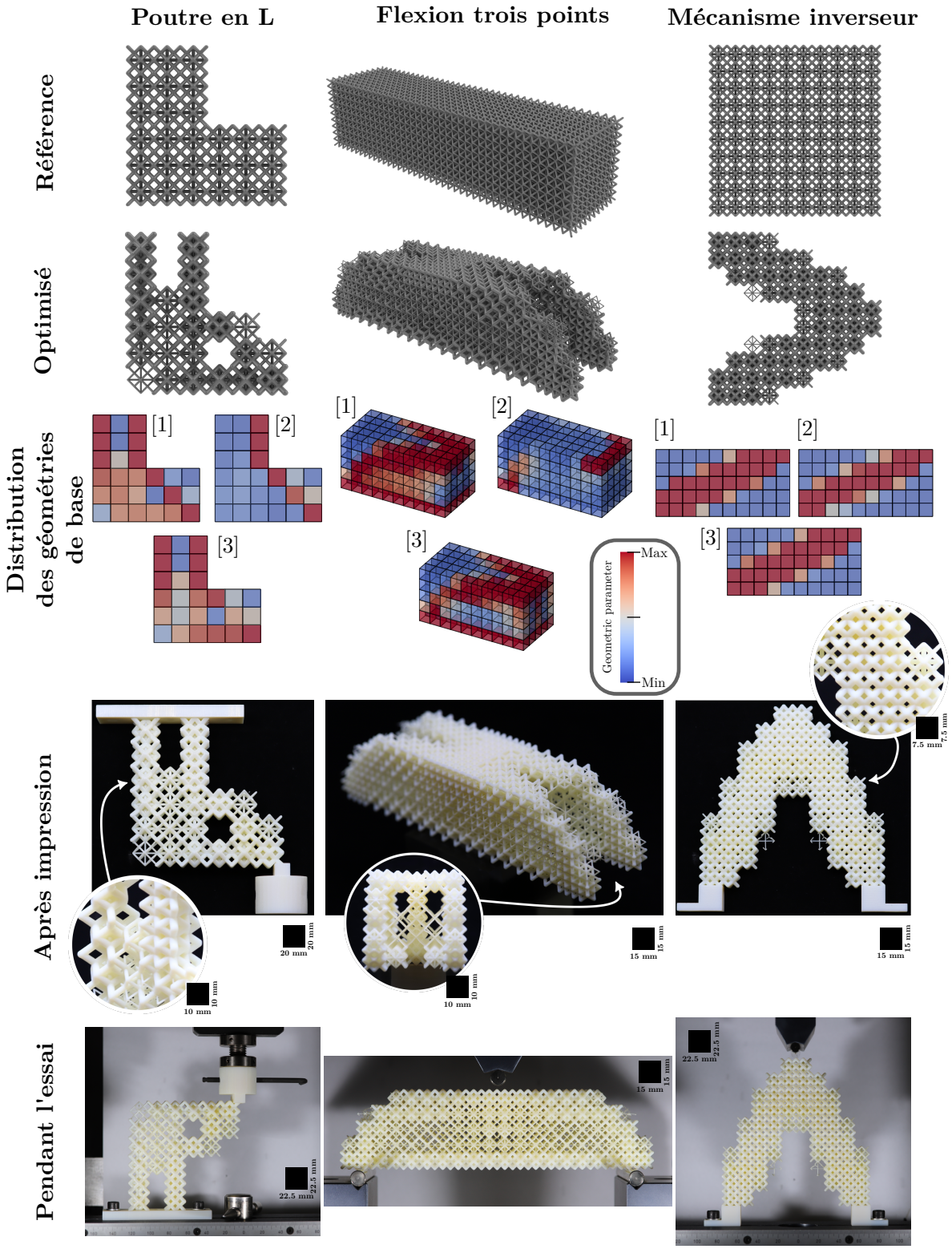

Fig. 7.13 Vue d'ensemble des cas tests expérimentaux comprenant les structures de référence et les conceptions optimisées. La ligne supérieure présente les modèles numériques de référence et optimisés pour les trois cas : poutre en L, flexion trois points et mécanisme inverseur. La ligne suivante illustre la distribution spatiale des géométries de base au sein des structures optimisées, où la couleur indique la valeur du paramètre géométrique (ici le rayon des poutres) variant du minimum (bleu) au maximum (rouge). Les images du bas montrent les spécimens imprimés par fabrication additive et leur configuration de test mécanique.

Les spécimens ont été fabriqués par procédé PolyJet sur une imprimante Stratasys Objet260 Connex3 en VeroWhite™, avec une épaisseur de couche de 30 μm et une résolution de 600 dpi. Un soin particulier a été apporté au post-traitement, comprenant un nettoyage à jet d'eau haute pression puis un bain d'hydroxyde de sodium pour éliminer le matériau de support résiduel. Cette étape s'est révélée critique pour garantir la fiabilité des essais mécaniques, bien que certaines difficultés de nettoyage aient pu endommager localement les structures les plus fines.

Les essais ont été réalisés sur une machine de traction Lloyd LR 50K équipée d'une cellule de charge de 50 kN, sous contrôle de déplacement à 2 mm/min. Les résultats expérimentaux montrent un bon accord avec les simulations numériques, les erreurs relatives sur la raideur variant de 0.6% à 25.1% selon le cas étudié. Les divergences observées proviennent principalement des imperfections d'impression, de la fragilisation lors du post-traitement et de la présence de matériau de support résiduel dans certaines cavités internes.

Globalement, les résultats confirment les tendances numériques avec des améliorations mécaniques substantielles pour les structures optimisées. Les gains de raideur atteignent 192% pour la poutre en L, 166% pour le mécanisme inverseur et jusqu'à 449% pour la poutre en flexion trois points. Ces observations démontrent la robustesse et l'efficacité du cadre d'optimisation développé, capable d'exploiter pleinement la variabilité géométrique locale pour concevoir des structures lattices légères et performantes, validées expérimentalement par la fabrication additive.

7.6 pyLattice : un outil de conception, de simulation et d'optimisation

Le développement d'un outil dédié à la conception, la simulation et l'optimisation de structures lattices treillis s'est révélé essentiel durant cette thèse. Le package Python `pyLattice`, accessible librement sur GitHub (https://github.com/Tcadart/pyLattice), a été conçu pour offrir à la communauté un environnement reproductible et extensible de modélisation numérique des structures lattices. Il permet la génération paramétrique de géométries, leur simulation par éléments finis et leur optimisation selon divers critères mécaniques, en s'appuyant sur les méthodologies développées dans ce manuscrit.

Conception de structures lattices. Le module de conception de `pyLattice` offre une grande flexibilité dans la génération de structures, avec la possibilité de choisir parmi

de nombreuses géométries (BCC, Diamond, Cubic, Octahedron, etc.) ou d'en combiner plusieurs pour créer des géométries hybrides. De multiples paramètres peuvent être définis : taille et nombre de cellules, rayon des poutres, gradients de propriétés, symétries, incertitudes de position des nœuds, ou encore blocs d'effacement pour créer des formes complexes. Les structures générées peuvent être exportées au format `STL`, morphed sur des fonctions géométriques ou intégrées dans des volumes solides, assurant ainsi la compatibilité directe avec la fabrication additive.

Simulation numérique. Le module de simulation, basé sur la bibliothèque `FEniCSX`, permet l'analyse mécanique de structures lattices à différentes échelles. Il intègre la méthode de pénalisation des poutres présentée à la Section 7.3, améliorant la précision de la modélisation des jonctions. Les possibilités incluent la simulation directe par éléments finis, la computation de propriétés homogénéisées des cellules unitaires, le calcul du complément de Schur, et la préparation à l'intégration future d'une méthode de décomposition de domaine. Les résultats peuvent être visualisés directement dans `matplotlib` ou être exportés vers ParaView™.

Optimisation des structures lattices. Le module d'optimisation de `pyLattice` permet la recherche de configurations optimales selon des objectifs tels que la minimisation de la compliance ou le contrôle de la densité relative. Plusieurs stratégies sont disponibles : optimisation à rayon constant, optimisation par gradients (linéaire, polynomiale, sinusoïdale) ou optimisation cellule par cellule avec paramètres indépendants. L'architecture du code, modulaire et extensible, permet d'intégrer facilement de nouveaux algorithmes ou formulations.

Conclusion. En résumé, `pyLattice` représente un environnement complet et ouvert pour la conception, la simulation et l'optimisation de structures lattices. Sa modularité, son intégration des méthodes de réduction de modèle et sa compatibilité avec des outils externes comme `Pyrough` en font une plateforme de référence pour la recherche et l'ingénierie de structures lattices légères, multifonctionnelles et adaptées à la fabrication additive.

7.7 Conclusion et perspectives

Cette thèse a présenté une méthodologie complète pour la conception et l'optimisation de structures lattices à géométrie variable, combinant la décomposition de domaine et les modèles de substitution afin de concilier précision et efficacité numérique. Après avoir proposé une méthode améliorée de pénalisation des poutres pour corriger le manque de rigidité aux jonctions, la recherche a introduit un cadre de simulation couplant la décomposition de domaine à des modèles réduits, le modèle de substitution à ordre réduit et le réseau de neurones augmenté par la physique, capables d'approximer efficacement le comportement local des cellules. Cette approche a ensuite été intégrée dans un cadre d'optimisation cellulaire, permettant d'explorer des espaces de conception de grande dimension à l'aide d'algorithmes à gradient. Les validations numériques et expérimentales ont démontré des gains significatifs en performance mécanique, tout en réduisant fortement les coûts de calcul.

Les perspectives de recherche se déclinent à plusieurs échelles temporelles : à court terme, l'enrichissement du modèle par des géométries plus complexes, des matériaux multiples et une meilleure prise en compte des contraintes liées à la fabrication additive ; à moyen terme, l'extension vers des comportements non linéaires et dynamiques grâce à des architectures de réseaux neuronaux plus avancées, telles que les réseaux de neurones en graphe ; et à long terme, la généralisation du cadre à des structures multi-physiques et multi-matériaux. Ainsi, cette thèse ouvre la voie à une conception numérique intégrée de structures lattices légères, adaptatives et optimisées pour des applications industrielles avancées.

References

Adams, D. and C. J. Turner (2018). "An Implicit Slicing Method for Additive Manufacturing Processes". In: *Virtual and Physical Prototyping* 13.1, pp. 2–7.

Adams, R., S. Townsend, S. Soe, and P. Theobald (2022). "Finite Element-Based Optimisation of an Elastomeric Honeycomb for Impact Mitigation in Helmet Liners". In: *International Journal of Mechanical Sciences* 214, p. 106920.

Adidas 4DFWD (2021). 3DNative. URL: https://www.3dnatives.com/adidas-4dfwd-3d-060520213/.

Ai, L. and X. Gao (2018). "Three-Dimensional Metamaterials with a Negative Poisson's Ratio and a Non-Positive Coefficient of Thermal Expansion". In: *International Journal of Mechanical Sciences* 135, pp. 101–113.

Al Ali, M. and M. Shimoda (2024). "Exploring the Influence of Initial Design Domain Dependencies in Concurrent Multiscale Topology Optimization for Heat Conductivity Maximization". In: *Computer Physics Communications* 295, p. 108968.

Al Rifaie, M., H. Abdulhadi, and A. Mian (2022). "Advances in Mechanical Metamaterials for Vibration Isolation: A Review". In: *Advances in Mechanical Engineering* 14.3, p. 16878132221082872.

Alaimo, A., A. Del Prete, G. Mantegna, C. Orlando, F. W. Panella, T. Primo, D. Tumino, and C. R. Vindigni (2023). "Modified Beam Modeling of Powder Bed Fusion Manufactured Lattice Structures". In: *International Journal of Mechanical Sciences* 259, p. 108599.

Ali, H. M. A. and M. Abdi (2023). "Multi-Objective Parametric Shape Optimisation of Body-Centred Cubic Lattice Structures for Additive Manufacturing". In: *Journal of Manufacturing and Materials Processing* 7.5 (5), p. 156.

Alkentar, R. and T. Mankovits (2025). "Development of Surrogate Model for Patient-Specific Lattice-Structured Hip Implant Design via Finite Element Analysis". In: *Applied Sciences* 15.7, p. 3522.

Almesmari, A., J. Sheikh-Ahmad, F. Jarrar, and S. Bojanampati (2023). "Optimizing the Specific Mechanical Properties of Lattice Structures Fabricated by Material Extrusion Additive Manufacturing". In: *Journal of Materials Research and Technology* 22, pp. 1821–1838.

Alnæs, M. S., A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells (2014). "Unified Form Language: A Domain-Specific Language for Weak Formulations of Partial Differential Equations". In: *ACM Transactions on Mathematical Software* 40.2, pp. 1–37.

Alzate Cobo, J. C., X.-L. Peng, B.-X. Xu, and O. Weeger (2025). *A Finite Swelling 3d Beam Model with Axial and Radial Diffusion*. URL: https://www.ssrn.com/abstract=5139774 (visited on 04/29/2025). Pre-published.

Amano, T., V. Ramírez-Castañeda, V. Berdejo-Espinola, I. Borokini, S. Chowdhury, M. Golivets, J. D. González-Trujillo, F. Montaño-Centellas, K. Paudel, R. L. White, and D. Veríssimo (2023). "The Manifold Costs of Being a Non-Native English Speaker in Science". In: *PLOS Biology* 21.7. Ed. by U. Dirnagl, e3002184.

André, J., A. Le Mehauté, and O. De Witte (1984). "Dispositif Pour Réaliser Un Modèle de Pièce Industrielle". French pat. 8411241.

Arabnejad, S. and D. Pasini (2013). "Mechanical Properties of Lattice Materials via Asymptotic Homogenization and Comparison with Alternative Homogenization Methods". In: *International Journal of Mechanical Sciences* 77, pp. 249–262.

Aremu, A., J. Brennan-Craddock, A. Panesar, I. Ashcroft, R. Hague, R. Wildman, and C. Tuck (2016). "A Voxel-Based Method of Constructing and Skinning Conformal and Functionally Graded Lattice Structures Suitable for Additive Manufacturing". In: *Additive Manufacturing* 13.

Arora, G., KiranBala, H. Emadifar, and M. Khademi (2023). "A Review of Radial Basis Function with Applications Explored". In: *Journal of the Egyptian Mathematical Society* 31.1, p. 6.

Arredondo-Soto, M., E. Cuan-Urquizo, and A. Gómez-Espinosa (2021). "A Review on Tailoring Stiffness in Compliant Systems, via Removing Material: Cellular Materials and Topology Optimization". In: *Applied Sciences* 11.8 (8), p. 3538.

Ashby, M. (2006). "The Properties of Foams and Lattices". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 364.1838, pp. 15–30.

Atkinson, K. E. (1989). *An Introduction to Numerical Analysis*. 2. ed. New York: Wiley. 693 pp.

Autodesk, Inc. (2025). *Fusion 360*.

Bai, R., H. Pu, G. Liang, H. Naceur, D. Coutellier, Y. Du, J. Zhao, J. Yi, X. Li, S. Yuan, J. Luo, and J. Lin (2024). "Exact Forming for Additive Manufacturing Using an Irregular Element-Based Compensating Approach: Simulation, Experiment, and Detection". In: *Mechanics of Advanced Materials and Structures* 31.26, pp. 7567–7578.

Baratta, I. A., J. P. Dean, J. S. Dokken, M. Habera, J. S. Hale, C. N. Richardson, M. E. Rognes, M. W. Scroggs, N. Sime, and G. N. Wells (2023). *DOLFINx: The next*

Generation FEniCS Problem Solving Environment. URL: https://zenodo.org/doi/10.5281/zenodo.10447666 (visited on 06/20/2025). Pre-published.

Bauchau, O. A. and J. I. Craig (2009). "Euler-Bernoulli Beam Theory". In: *Structural Analysis*. Ed. by O. A. Bauchau and J. I. Craig. Red. by G. M. L. Gladwell. Vol. 163. Dordrecht: Springer Netherlands, pp. 173–221.

Beevers, E., A. Cutolo, and B. Van Hooreweder (2025). "Finetuning Design and Process Conditions to Minimise Manufacturing Deviations in Strut-Based Metal Lattice Structures Produced by Laser Powder Bed Fusion". In: *Applied Materials Today* 42, p. 102551.

Bhattacharyya, S., J. Tao, E. Gildin, and J. C. Ragusa (2025). "Hyper-Reduction Techniques for Efficient Simulation of Large-Scale Engineering Systems". In: *Archives of Computational Methods in Engineering*.

Boursier Niutta, C., R. Ciardiello, and A. Tridello (2022). "Experimental and Numerical Investigation of a Lattice Structure for Energy Absorption: Application to the Design of an Automotive Crash Absorber". In: *Polymers* 14, p. 1116.

Brent, R. P. and T. J. Watson (1973). *Algorithms for Minimization without Derivatives*. George Forsythe. Prentice-Hall.

Brivio, S., S. Fresca, and A. Manzoni (2024). "PTPI-DL-ROMs: Pre-trained Physics-Informed Deep Learning-Based Reduced Order Models for Nonlinear Parametrized PDEs". In: *Computer Methods in Applied Mechanics and Engineering* 432, p. 117404.

Burman, E., S. Claus, P. Hansbo, M. G. Larson, and A. Massing (2015). "CutFEM: Discretizing Geometry and Partial Differential Equations". In: *International Journal for Numerical Methods in Engineering* 104.7, pp. 472–501.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025a). "Conception numérique de structures lattice par décomposition de domaine et réseau neuronal". In: *19ème colloque national S.mart*. Vogüé, Ardèche, France.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025b). "Optimal Design of Lattice Structures via PINN-based Domain Decomposition". In: *The Fifth International Conference on Simulation for Additive Manufacturing*. Pavia, Italy.

Cadart, T., T. Hirschler, S. Bahi, S. Roth, F. Demoly, and N. Lebaal (2025). "An Optimal Penalty Method for the Joint Stiffening in Beam Models of Additively Manufactured Lattice Structures". In: *International Journal of Solids and Structures* 306, p. 113107.

Cadart, T., T. Hirschler, S. Bahi, S. Roth, F. Demoly, and N. Lebaal (2024). "Une modélisation poutre enrichie des structures lattices en fabrication additive". In: *16e colloque national en calcul des structures*. Giens, Var, France.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (2025c). "A PINN-based Domain Decomposition Method for the Optimal Design of Additively Manufactured Lattice Structures". In: *12th European Solid Mechanics Conference*. Lyon, France.

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (n.d.[a]). "A PANN-enhanced Domain Decomposition Method for the Optimal Design of Additively Manufactured Lattice Structures". In: *preparation* ().

Cadart, T., T. Hirschler, S. Roth, F. Demoly, S. Bahi, and N. Lebaal (n.d.[b]). "pyLattice: An Open-Source Python Package for Design and Simulation of Lattice Structures". In: *preparation* ().

Cai, S., Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis (2021). "Physics-Informed Neural Networks (PINNs) for Fluid Mechanics: A Review". In: *Acta Mechanica Sinica* 37.12, pp. 1727–1738.

Carbon, Inc. (2025). *Carbon Design Engine*.

Chami, I., S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy (2022). *Machine Learning on Graphs: A Model and Comprehensive Taxonomy*. URL: http://arxiv.org/abs/2005.03675 (visited on 12/15/2023). Pre-published.

Chan–Renous-Legoubin, R. and C. W. Royer (2022). *A Nonlinear Conjugate Gradient Method with Complexity Guarantees and Its Application to Nonconvex Regression*. URL: http://arxiv.org/abs/2201.08568 (visited on 09/18/2025). Pre-published.

Chen, J., M. Baldwin, S. Prabha Narra, and C. McComb (2025). "Multi-Lattice Topology Optimization Via Generative Lattice Modeling". In: *Journal of Mechanical Design* 147.5, p. 051707.

Chen, J., R. Yang, R. Ma, and J. Li (2016). "Design Optimization of Wind Turbine Tower with Lattice-Tubular Hybrid Structure Using Particle Swarm Algorithm: Design Optimization of Lattice-tubular Wind Turbine Tower". In: *The Structural Design of Tall and Special Buildings* 25.15, pp. 743–758.

Chen, Z., Z. Wang, S. Zhou, J. Shao, and X. Wu (2018). "Novel Negative Poisson's Ratio Lattice Structures with Enhanced Stiffness and Energy Absorption Capacity". In: *Materials* 11.7, p. 1095.

Chougrani, L., J.-P. Pernot, P. Véron, and S. Abed (2017). "Lattice Structure Lightweight Triangulation for Additive Manufacturing". In: *Computer-Aided Design* 90, pp. 95–104.

Chua, J. W., X. Li, X. Yu, and W. Zhai (2023). "Novel Slow-Sound Lattice Absorbers Based on the Sonic Black Hole". In: *Composite Structures* 304, p. 116434.

Costa, M. R., A. Sohouli, and A. Suleman (2022). "Multi-Scale and Multi-Material Topology Optimization of Gradient Lattice Structures Using Surrogate Models". In: *Composite Structures* 289, p. 115402.

Dadashi, A. and G. H. Rahimi (2025). "A Novel Analytical Solution for Determining the Shear Behavior of the Lattice Structure Based on the Primitive Unit Cell Considering the Shear and Bending Effect". In: *Mechanics Based Design of Structures and Machines* 53.1, pp. 219–244.

Dal Fabbro, P., S. Rosso, A. Ceruti, D. Boscolo Bozza, R. Meneghello, G. Concheri, and G. Savio (2021). "Analysis of a Preliminary Design Approach for Conformal Lattice Structures". In: *Applied Sciences* 11.23, p. 11449.

Damanpack, A. R., M. Bodaghi, and W. H. Liao (2019). "Experimentally Validated Multi-Scale Modeling of 3D Printed Hyper-Elastic Lattices". In: *International Journal of Non-Linear Mechanics* 108, pp. 87–110.

Daniel, T., F. Casenave, N. Akkari, and D. Ryckelynck (2020). "Model Order Reduction Assisted by Deep Neural Networks (ROM-net)". In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1, p. 16.

De Biasi, R., M. Salman Yasin, M. Perini, M. Benedetti, and F. Berto (2025). "Compensated Beam Model for Efficient and Accurate FE Elastic Simulation of Strut-Based Lattice Structures". In: *Materials & Design*, p. 114213.

Delahaye, D., S. Chaimatanan, and M. Mongeau (2019). "Simulated Annealing: From Basics to Applications". In: *Handbook of Metaheuristics*. Ed. by M. Gendreau and J.-Y. Potvin. Vol. 272. Cham: Springer International Publishing, pp. 1–35.

Dohrmann, C. R. (2003). "A Preconditioner for Substructuring Based on Constrained Energy Minimization". In: *SIAM Journal on Scientific Computing* 25.1, pp. 246–258.

Dold, D. and D. Aranguren Van Egmond (2023). "Differentiable Graph-Structured Models for Inverse Design of Lattice Materials". In: *Cell Reports Physical Science* 4.10, p. 101586.

Dong, G., G. Wijaya, Y. Tang, and Y. F. Zhao (2018). "Optimizing Process Parameters of Fused Deposition Modeling by Taguchi Method for the Fabrication of Lattice Structures". In: *Additive Manufacturing* 19, pp. 62–72.

Donnici, G., M. Freddi, and A. Liverani (2024). "RSM Applied to Lattice Patterns for Stiffness Optimization". In: *Rapid Prototyping Journal* 30.11, pp. 345–356.

Dos Reis, F. and N. Karathanasopoulos (2022). "Inverse Metamaterial Design Combining Genetic Algorithms with Asymptotic Homogenization Schemes". In: *International Journal of Solids and Structures* 250, p. 111702.

Dowd, P. A. and E. Pardo-Igúzquiza (2024). "The Many Forms of Co-kriging: A Diversity of Multivariate Spatial Estimators". In: *Mathematical Geosciences* 56.2, pp. 387–413.

Druschitz, A. P., S. Cowden, A. Dudley, S. Walsh, A. Weir, C. B. Williams, and B. Wood (2016). "Metal-Ceramic Composite Lattice Structures Using 3D Printed Sand Molds and Cores". In: *2016 International Solid Freeform Fabrication Symposium*. University of Texas at Austin.

Duan, C. and D. Wu (2025). "Inverse Design of Lattice Structures with Target Mechanical Performance via Generative Adversarial Networks Considering the Effect of Process Parameters". In: *Advanced Engineering Informatics* 65, p. 103221.

Duncan, O., T. Shepherd, C. Moroney, L. Foster, P. D. Venkatraman, K. Winwood, T. Allen, and A. Alderson (2018). "Review of Auxetic Materials for Sports Applications: Expanding Options in Comfort and Protection". In: *Applied Sciences* 8.6, p. 941.

Dwyer, C. M., J. G. Carrillo, J. A. D. De la Peña, C. C. Santiago, E. MacDonald, J. Rhinehart, R. M. Williams, M. Burhop, B. Yelamanchi, and P. Cortes (2023). "Impact Performance of 3D Printed Spatially Varying Elastomeric Lattices". In: *Polymers* 15.5, p. 1178.

Echeta, I., B. Dutton, R. K. Leach, and S. Piano (2021). "Finite Element Modelling of Defects in Additively Manufactured Strut-Based Lattice Structures". In: *Additive Manufacturing* 47, p. 102301.

Echeta, I., X. Feng, B. Dutton, R. Leach, and S. Piano (2020). "Review of Defects in Lattice Structures Manufactured by Powder Bed Fusion". In: *The International Journal of Advanced Manufacturing Technology* 106.5, pp. 2649–2668.

Eigen, D., M. Ranzato, and I. Sutskever (2014). *Learning Factored Representations in a Deep Mixture of Experts*. URL: http://arxiv.org/abs/1312.4314 (visited on 09/18/2025). Pre-published.

Farhat, C., M. Lesoinne, and K. Pierson (2000). "A Scalable Dual-Primal Domain Decomposition Method". In: *Numerical Linear Algebra with Applications* 7.7–8, pp. 687–714.

Farhat, C. and F.-X. Roux (1991). "A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm". In: *International Journal for Numerical Methods in Engineering* 32.6, pp. 1205–1227.

Ferro, N., S. Perotto, and M. Gavazzoni (2022). "A New Fluid-Based Strategy for the Connection of Non-Matching Lattice Materials". In: *Structural and Multidisciplinary Optimization* 65.10, p. 287.

Fizik 3D (2022). 3DNative. URL: https://www.3dnatives.com/en/fizik-3d-saddles-150420224/.

Fresca, S., L. Dede', and A. Manzoni (2021). "A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs". In: *Journal of Scientific Computing* 87.2, p. 61.

Fresca, S. and A. Manzoni (2022). "POD-DL-ROM: Enhancing Deep Learning-Based Reduced Order Models for Nonlinear Parametrized PDEs by Proper Orthogonal Decomposition". In: *Computer Methods in Applied Mechanics and Engineering* 388, p. 114181.

Frey, R., M. R. Tucker, M. Afrasiabi, and M. Bambach (2025). "Multi-Objective Design of Multi-Material Truss Lattices Utilizing Graph Neural Networks". In: *Scientific Reports* 15.1, p. 3187.

Frisch, K. C. (1981). "History of Science and Technology of Polymeric Foams". In: *Journal of Macromolecular Science: Part A - Chemistry* 15.6, pp. 1089–1112.

Fu, J., L. Zhang, X. Song, and M. Fu (n.d.). "Effect of Shell Thickness on the Deformation Mode of TPMS Lattices in Quasi-Static Compression". In: ().

Gao, J., M. Xiao, Y. Zhang, and L. Gao (2020). "A Comprehensive Review of Isogeometric Topology Optimization: Methods, Applications and Prospects". In: *Chinese Journal of Mechanical Engineering* 33.1, p. 87.

Gao, Z., H. Wang, N. Letov, Y. F. Zhao, X. Zhang, Y. Wu, C. L. A. Leung, and H. Wang (2023). "Data-Driven Design of Biometric Composite Metamaterials with Extremely Recoverable and Ultrahigh Specific Energy Absorption". In: *Composites Part B: Engineering* 251, p. 110468.

Gärtner, T., M. Fernández, and O. Weeger (2021). "Nonlinear Multiscale Simulation of Elastic Beam Lattices with Anisotropic Homogenized Constitutive Models Based on Artificial Neural Networks". In: *Computational Mechanics* 68.5, pp. 1111–1130.

Gebisa, A. W. and H. G. Lemu (2017). "A Case Study on Topology Optimized Design for Additive Manufacturing". In: *IOP Conference Series: Materials Science and Engineering* 276, p. 012026.

Geuzaine, C. and J.-F. Remacle (2009). "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities". In: *International Journal for Numerical Methods in Engineering* 79.11, pp. 1309–1331.

Gibson, I., D. Rosen, and B. Stucker (2015). *Additive Manufacturing Technologies*. New York, NY: Springer New York.

Gibson, L. J. and M. F. Ashby (1997). *Cellular Solids: Structure and Properties*. 2nd ed. Cambridge University Press.

Glaesener, R. N., J.-H. Bastek, F. Gonon, V. Kannan, B. Telgen, B. Spöttling, S. Steiner, and D. M. Kochmann (2021). "Viscoelastic Truss Metamaterials as Time-Dependent Generalized Continua". In: *Journal of the Mechanics and Physics of Solids* 156, p. 104569.

Glaesener, R. N., C. Lestringant, B. Telgen, and D. M. Kochmann (2019). "Continuum Models for Stretching- and Bending-Dominated Periodic Trusses Undergoing Finite Deformations". In: *International Journal of Solids and Structures* 171, pp. 117–134.

Glaesener, R. N., E. A. Träff, B. Telgen, R. M. Canonica, and D. M. Kochmann (2020). "Continuum Representation of Nonlinear Three-Dimensional Periodic Truss Networks by on-the-Fly Homogenization". In: *International Journal of Solids and Structures* 206, pp. 101–113.

Glorot, X. and Y. Bengio (2010). "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In.

Gómez-Silva, F., R. Zaera, R. Ortigosa, and J. Martínez-Frutos (2025). "Topology Optimization of Lattice Structures for Target Band Gaps with Optimum Volume Fraction via Bloch-Floquet Theory". In: *Computers & Structures* 307, p. 107601.

Gongora, A. E., C. Friedman, D. K. Newton, T. D. Yee, Z. Doorenbos, B. Giera, E. B. Duoss, T. Y.-J. Han, K. Sullivan, and J. N. Rodriguez (2024). "Accelerating the Design of Lattice Structures Using Machine Learning". In: *Scientific Reports* 14.1, p. 13703.

Gosselet, P. and C. Rey (2006). "Non-Overlapping Domain Decomposition Methods in Structural Mechanics". In: *Archives of Computational Methods in Engineering* 13.4, pp. 515–572.

Groen, J. P. and O. Sigmund (2018). "Homogenization-based Topology Optimization for High-resolution Manufacturable Microstructures". In: *International Journal for Numerical Methods in Engineering* 113.8, pp. 1148–1163.

Guerder, M., A. Duval, T. Elguedj, and D. Dam (2023). "Volumetric Embedded Entities for the IsoGeometric Analysis of Complex Structures". In: *Computer Methods in Applied Mechanics and Engineering* 417, p. 116426.

Gülcan, O., K. Günaydın, and A. Tamer (2021). "The State of the Art of Material Jetting—A Critical Review". In: *Polymers* 13.16, p. 2829.

Gulkanov, A. and K. Modestov (2019). "Analytical Methods of Theoretical Physics in the Modeling of Building Structures". In: *Journal of Physics: Conference Series* 1425.1, p. 012180.

Gümrük, R. and R. A. W. Mines (2013). "Compressive Behaviour of Stainless Steel Micro-Lattice Structures". In: *International Journal of Mechanical Sciences* 68, pp. 125–139.

Guo, H., A. Takezawa, M. Honda, C. Kawamura, and M. Kitamura (2020). "Finite Element Simulation of the Compressive Response of Additively Manufactured Lattice Structures with Large Diameters". In: *Computational Materials Science* 175, p. 109610.

Guo, W., J. Li, W. Luo, J. Yang, X. Zhu, and J. Yan (2025). "A New Method for Solving Parameter Mutation Analysis in Periodic Structure Bandgap Calculation". In: *European Journal of Mechanics - A/Solids* 111, p. 105572.

Ha, N. S. and G. Lu (2020). "A Review of Recent Research on Bio-Inspired Structures and Materials for Energy Absorption Applications". In: *Composites Part B: Engineering* 181, p. 107496.

Hager, W. W. and H. Zhang (2006). "A SURVEY OF NONLINEAR CONJUGATE GRADIENT METHODS". In.

Harris, J. A. and G. J. McShane (2021). "Impact Response of Metallic Stacked Origami Cellular Materials". In: *International Journal of Impact Engineering* 147, p. 103730.

Hassani, B. and E. Hinton (1998). "A Review of Homogenization and Topology Opimization II—Analytical and Numerical Solution of Homogenization Equations". In: *Computers & Structures* 69.6, pp. 719–738.

He, J., S. Koric, S. Kushwaha, J. Park, D. Abueidda, and I. Jasiuk (2023). "Novel DeepONet Architecture to Predict Stresses in Elastoplastic Structures with Variable

Complex Geometries and Loads". In: *Computer Methods in Applied Mechanics and Engineering* 415, p. 116277.

Henyš, P., L. Čapek, and J. Březina (2019). "Comparison of Current Methods for Implementing Periodic Boundary Conditions in Multi-Scale Homogenisation". In: *European Journal of Mechanics - A/Solids* 78, p. 103825.

Herrmann, L. and S. Kollmannsberger (2024). "Deep Learning in Computational Mechanics: A Review". In: *Computational Mechanics*.

Hess, M. W., A. Quaini, and G. Rozza (2021). "A Comparison of Reduced-Order Modeling Approaches Using Artificial Neural Networks for PDEs with Bifurcating Solutions". In: *ETNA - Electronic Transactions on Numerical Analysis* 56, pp. 52–65.

Hestenes, M. R. and E. Stiefel (1952). "Methods of Conjugate Gradients for Solving Linear Systems". In.

Hesthaven, J. S., G. Rozza, and B. Stamm (2016). *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Cham: Springer International Publishing.

Hirschler, T., P. Antolin, and A. Buffa (2022). "Fast and Multiscale Formation of Isogeometric Matrices of Microstructured Geometric Models". In: *Computational Mechanics* 69.2, pp. 439–466.

Hirschler, T., R. Bouclier, P. Antolin, and A. Buffa (2024). "Reduced Order Modeling Based Inexact FETI-DP Solver for Lattice Structures". In: *International Journal for Numerical Methods in Engineering* 125.8, e7419.

Hoang, V.-N., T. Pham, D. Ho, and H. Nguyen-Xuan (2022). "Robust Multiscale Design of Incompressible Multi-Materials under Loading Uncertainties". In: *Engineering with Computers* 38.1, pp. 875–890.

Hoang, V.-N., P. Tran, V.-T. Vu, and H. Nguyen-Xuan (2020). "Design of Lattice Structures with Direct Multiscale Topology Optimization". In: *Composite Structures* 252, p. 112718.

Hössinger-Kalteis, A., M. Reiter, M. Jerabek, and Z. Major (2021). "Overview and Comparison of Modelling Methods for Foams". In: *Journal of Cellular Plastics* 57.6, pp. 951–1001.

Hu, H., L. Qi, and X. Chao (2024). "Physics-Informed Neural Networks (PINN) for Computational Solid Mechanics: Numerical Frameworks and Applications". In: *Thin-Walled Structures* 205, p. 112495.

Hu, Q., G. Lu, and K. M. Tse (2023). "Compressive and Tensile Behaviours of 3D Hybrid Auxetic-Honeycomb Lattice Structures". In: *International Journal of Mechanical Sciences*, p. 108767.

Iteney, H., J. A. Gonzalez Joa, C. Le Bourlot, T. W. Cornelius, O. Thomas, and J. Amodeo (2024). "Pyrough: A Tool to Build 3D Samples with Rough Surfaces for

Atomistic and Finite-Element Simulations". In: *Computer Physics Communications* 295, p. 108958.

Jain, A., E. Haghighat, and S. Nelaturi (2024). *LatticeGraphNet: A Two-Scale Graph Neural Operator for Simulating Lattice Structures*. URL: http://arxiv.org/abs/2402.01045 (visited on 07/11/2024). Pre-published.

Jaisawal, R. and V. Agrawal (2021). "Generative Design Method (GDM) – A State of Art". In: *IOP Conference Series: Materials Science and Engineering* 1104.1, p. 012036.

Jalali, S. K., M. J. Beigrezaee, D. Misseroni, and N. M. Pugno (2024). "A Modified Gibson-Ashby Model for Functionally Graded Lattice Structures". In: *Mechanics of Materials* 188, p. 104822.

Jansari, C., S. P. Bordas, M. Montemurro, and E. Atroshchenko (2025). "Design of Thermal Meta-Structures Made of Functionally Graded Materials Using Isogeometric Density-Based Topology Optimization". In: *Composite Structures* 364, p. 119114.

JAXA Moon Lander (2024). Metal AM. URL: https://www.metal-am.com/japans-slim-lands-on-moon-with-help-from-metal-additive-manufacturing/.

Ji, W., Y. Bai, C. Jiang, J. Liu, Q. Yan, and X. Han (2024). "Concurrent Topology Optimization of Sandwich Structures with Multi-Configuration and Variable-Diameter Lattice Infill". In: *Computer Methods in Applied Mechanics and Engineering* 432, p. 117324.

Jung, J., G. Meyer, M. Greiner, and C. Mittelstedt (2023). "Load Introduction Specimen Design for the Mechanical Characterisation of Lattice Structures under Tensile Loading". In: *Journal of Manufacturing and Materials Processing* 7.1 (1), p. 37.

Karimi, S. and S. Vavasis (2024). *Nonlinear Conjugate Gradient for Smooth Convex Functions*. URL: http://arxiv.org/abs/2111.11613 (visited on 09/18/2025). Pre-published.

Katoch, S., S. S. Chauhan, and V. Kumar (2021). "A Review on Genetic Algorithm: Past, Present, and Future". In: *Multimedia Tools and Applications* 80.5, pp. 8091–8126.

Kazemi, H., A. Vaziri, and J. A. Norato (2020). "Multi-Material Topology Optimization of Lattice Structures Using Geometry Projection". In: *Computer Methods in Applied Mechanics and Engineering* 363, p. 112895.

Kerfriden, P., S. Claus, and I. Mihai (2020). "A Mixed-Dimensional CutFEM Methodology for the Simulation of Fibre-Reinforced Composites". In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1, p. 18.

Khan, N. and A. Riccio (2024). "A Systematic Review of Design for Additive Manufacturing of Aerospace Lattice Structures: Current Trends and Future Directions". In: *Progress in Aerospace Sciences* 149, p. 101021.

Khuri, A. I. and S. Mukhopadhyay (2010). "Response Surface Methodology". In: *WIREs Computational Statistics* 2.2, pp. 128–149.

Kim, W. and H. Choi (2019). "Immersed Boundary Methods for Fluid-Structure Interaction: A Review". In: *International Journal of Heat and Fluid Flow* 75, pp. 301–309.

Kircher, T., F. Döppel, and M. Votsmeier (2023). *A Neural Network with Embedded Stoichiometry and Thermodynamics for Learning Kinetics from Reactor Data*. URL: https://chemrxiv.org/engage/chemrxiv/article-details/650070e8b6ab98a41c5213b5 (visited on 09/18/2025). Pre-published.

Klawonn, A. and O. Rheinbach (2006). "A Parallel Implementation of Dual-Primal FETI Methods for Three-Dimensional Linear Elasticity Using a Transformation of Basis". In: *SIAM Journal on Scientific Computing* 28.5, pp. 1886–1906.

Kleijnen, J. P. (2009). "Kriging Metamodeling in Simulation: A Review". In: *European Journal of Operational Research* 192.3, pp. 707–716.

Klein, D. K., R. Ortigosa, J. Martínez-Frutos, and O. Weeger (2022). "Finite Electro-Elasticity with Physics-Augmented Neural Networks". In: *Computer Methods in Applied Mechanics and Engineering* 400, p. 115501.

Koduru, J. P., K. L. Narayana, and K. M. Mantrala (2022). "Hybrid Swarm-Based Intelligent Algorithm for Lattice Structure Optimization in Additive Manufacturing System". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 16.4, pp. 1511–1524.

Köll, J. and S. Hallström (2016). "Elastic Properties of Equilibrium Foams". In: *Acta Materialia* 113, pp. 11–18.

Korshunova, N., G. Alaimo, S. B. Hosseini, M. Carraturo, A. Reali, J. Niiranen, F. Auricchio, E. Rank, and S. Kollmannsberger (2021). "Image-Based Numerical Characterization and Experimental Validation of Tensile Behavior of Octet-Truss Lattice Structures". In: *Additive Manufacturing* 41, p. 101949.

Korshunova, N., I. Papaioannou, S. Kollmannsberger, D. Straub, and E. Rank (2021). "Uncertainty Quantification of Microstructure Variability and Mechanical Behavior of Additively Manufactured Lattice Structures". In: *Computer Methods in Applied Mechanics and Engineering* 385, p. 114049.

Kothandaraman, L. and N. K. Balasubramanian (2024). "Optimization of FDM Printing Parameters for Square Lattice Structures: Improving Mechanical Characteristics". In: *Materials Today: Proceedings*, S2214785324002219.

Labeas, G. N. and M. M. Sunaric (2010). "Investigation on the Static Response and Failure Process of Metallic Open Lattice Cellular Structures". In: *Strain* 46.2, pp. 195–204.

Lattice Medical (2023). URL: https://www.lattice-medical.com/.

Leiden, U. (2015). *Thunthu - 3D Model Bones of the Lower Limb*.

Li, D., W. Liao, N. Dai, and Y. M. Xie (2020). "Anisotropic Design and Optimization of Conformal Gradient Lattice Structures". In: *Computer-Aided Design* 119, p. 102787.

Li, D., R. Qin, B. Chen, and J. Zhou (2021). "Analysis of Mechanical Properties of Lattice Structures with Stochastic Geometric Defects in Additive Manufacturing". In: *Materials Science and Engineering: A* 822, p. 141666.

Li, D., R. Qin, J. Xu, J. Zhou, and B. Chen (2022). "Topology Optimization of Thin-Walled Tubes Filled with Lattice Structures". In: *International Journal of Mechanical Sciences* 227, p. 107457.

Li, L., J. Wu, F. Yang, P. Li, G. Lu, H. Fan, J. Zhao, N. Zhang, and Q. Yang (2025). "Mechanisms of Low-Frequency Bandgap Formation and Energy Absorption of Three-Dimensional Nested Hybrid Lattice Structures". In: *Composites Part B: Engineering* 291, p. 112045.

Li, Y., Z. Feng, L. Hao, L. Huang, C. Xin, Y. Wang, E. Bilotti, K. Essa, H. Zhang, Z. Li, F. Yan, and T. Peijs (2020). "A Review on Functionally Graded Materials and Structures via Additive Manufacturing: From Multi-Scale Design to Versatile Functional Properties". In: *Advanced Materials Technologies* 5.6, p. 1900981.

Li, Z., D. Z. Huang, and B. Liu (n.d.). "Fourier Neural Operator with Learned Deformations for PDEs on General Geometries". In: ().

Ling, C., A. Cernicchi, M. D. Gilchrist, and P. Cardiff (2019). "Mechanical Behaviour of Additively-Manufactured Polymeric Octet-Truss Lattice Structures under Quasi-Static and Dynamic Compressive Loading". In: *Materials & Design* 162, pp. 106–118.

Liu, C. and S. Li (2023). "High-Resolution Topology Optimization Method of Multi-Morphology Lattice Structures Based on Three-Dimensional Convolutional Neural Networks (3D-CNN)". In: *Structural and Multidisciplinary Optimization* 66.11, p. 235.

Liu, D., Y. Zhao, S. Zhang, M. Li, and G. Pang (2025). "Superior Mechanical Properties of an Additively Manufactured Gradient Lattice Structure through FEM Simulation and In-Situ Compression Tests". In: *Materials Today Communications* 42, p. 111115.

Liu, F., L. Wang, D. Jin, X. Liu, and P. Lu (2021). "Equivalent Beam Model for Spatial Repetitive Lattice Structures with Hysteretic Nonlinear Joints". In: *International Journal of Mechanical Sciences* 200, p. 106449.

Liu, F., M. Chen, L. Wang, Z. Xiang, and S. Huang (2024). "Lightweight and Customized Design via Conformal Parametric Lattice Driven by Stress Fields". In: *Phygital Intelligence*. Ed. by C. Yan, H. Chai, T. Sun, and P. F. Yuan. Singapore: Springer Nature, pp. 139–149.

Liu, H., C. Qiao, Y. Ma, F. Pan, and Y. Chen (2023). "A Concave-Convex Design Strategy for Periodic Self-Locked Energy-Absorbing Systems". In: *International Journal of Impact Engineering* 175, p. 104539.

Liu, H., W. W. S. Ma, J. Ding, S. Qu, R. Li, Q. Ge, M. Y. Wang, and X. Song (2024). "Hierarchical Triply Periodic Minimal Surface Shell Lattices with Superior Isotropic Elasticity: Design Guidelines, Fabrication, and Validation". In: *Additive Manufacturing* 94, p. 104451.

Liu, J., K. Guo, J. Sun, Q. Sun, L. Wang, and H. Li (2021). "Compressive Behavior and Vibration-Damping Properties of Porous Ti-6Al-4V Alloy Manufactured by Laser Powder Bed Fusion". In: *Journal of Manufacturing Processes* 66, pp. 1–10.

Liu, J., Z. Zou, K. Gao, J. Yang, S. He, and Z. Wu (2025). "A Novel Digital Unit Cell Library Generation Framework for Topology Optimization of Multi-Morphology Lattice Structures". In: *Composite Structures* 354, p. 118824.

Liu, S., T. Liu, Q. Zou, W. Wang, E. L. Doubrovski, and C. C. L. Wang (2021). *Memory-Efficient Modeling and Slicing of Large-Scale Adaptive Lattice Structures*. URL: http://arxiv.org/abs/2101.05031 (visited on 04/17/2025). Pre-published.

Liu, X., L. Gao, M. Xiao, and Y. Zhang (2022). "Kriging-Assisted Design of Functionally Graded Cellular Structures with Smoothly-Varying Lattice Unit Cells". In: *Computer Methods in Applied Mechanics and Engineering* 390, p. 114466.

Liu, Y., J. He, and J. Liang (n.d.). "A New Improved Particle Swarm Algorithm for Optimization of Anchor Lattice Beam Support Structures". In: ().

Liu, Y., P. Zhang, W. Xu, W. Zeng, Y.-J. Yang, and W. Wang (2025). "Design and Optimization of Variable Radii Self-Supporting Lattice Structures". In: *Computer Methods in Applied Mechanics and Engineering* 434, p. 117510.

Liu, Y., H. Yang, Y. F. Zhao, and G. Zheng (2021). "A Heterogeneous Lattice Structure Modeling Technique Supported by Multiquadric Radial Basis Function Networks". In: *Journal of Computational Design and Engineering* 9.1, pp. 68–81.

Loshchilov, I. and F. Hutter (2019). *Decoupled Weight Decay Regularization*. URL: http://arxiv.org/abs/1711.05101 (visited on 09/16/2025). Pre-published.

Lozanovski, B., D. Downing, R. Tino, A. du Plessis, P. Tran, J. Jakeman, D. Shidid, C. Emmelmann, M. Qian, P. Choong, M. Brandt, and M. Leary (2020). "Non-Destructive Simulation of Node Defects in Additively Manufactured Lattice Structures". In: *Additive Manufacturing* 36, p. 101593.

Lozanovski, B., D. Downing, P. Tran, D. Shidid, M. Qian, P. Choong, M. Brandt, and M. Leary (2020). "A Monte Carlo Simulation-Based Approach to Realistic Modelling of Additively Manufactured Lattice Structures". In: *Additive Manufacturing* 32, p. 101092.

Luxner, M. H., J. Stampfl, and H. E. Pettermann (2005). "Finite Element Modeling Concepts and Linear Analyses of 3D Regular Open Cell Structures". In: *Journal of Materials Science* 40.22, pp. 5859–5866.

Ma, J., H. Zhang, T.-U. Lee, H. Lu, Y. M. Xie, and N. S. Ha (2025). "Auxetic Behavior and Energy Absorption Characteristics of a Lattice Structure Inspired by Deep-Sea Sponge". In: *Composite Structures* 354, p. 118835.

Ma, Q., L. Zhang, J. Ding, S. Qu, J. Fu, M. Zhou, M. W. Fu, X. Song, and M. Y. Wang (2021). "Elastically-Isotropic Open-Cell Minimal Surface Shell Lattices with Superior Stiffness via Variable Thickness Design". In: *Additive Manufacturing* 47, p. 102293.

Maday, Y. and F. Magoulès (2006). "Absorbing Interface Conditions for Domain Decomposition Methods: A General Presentation". In: *Computer Methods in Applied Mechanics and Engineering* 195.29–32, pp. 3880–3900.

Mandel, J. (1993). "Balancing Domain Decomposition". In: *Communications in Numerical Methods in Engineering* 9.3, pp. 233–241.

Mandel, J., C. R. Dohrmann, and R. Tezaur (2005). "An Algebraic Theory for Primal and Dual Substructuring Methods by Constraints". In: *Applied Numerical Mathematics* 54.2, pp. 167–193.

Marchal, V., T. Shujie, Y. Zhang, N. Labed, and F. Peyraut (2024). "Multiscale Periodic Homogenization for Additive Manufacturing of Honeycomb Lattices". In: *International Journal of Solids and Structures* 302, p. 112974.

Margossian, C. C. (2019). "A Review of Automatic Differentiation and Its Efficient Implementation". In: *WIREs Data Mining and Knowledge Discovery* 9.4, e1305.

Maurizi, M., C. Gao, and F. Berto (2022). "Predicting Stress, Strain and Deformation Fields in Materials and Structures with Graph Neural Networks". In: *Scientific Reports* 12.1 (1), p. 21834.

Maxwell, J. C. (1864). "L. *On the Calculation of the Equilibrium and Stiffness of Frames*". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 27.182, pp. 294–299.

McCulloch, W. S. and W. Pitts (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The Bulletin of Mathematical Biophysics* 5.4, pp. 115–133.

McDonnell, B., E. M. O'Hara, and N. M. Harrison (2024). "Simulation-Driven-Design of Metal Lattice Structures for a Target Stress–Strain Curve". In: *Materials & Design* 237, p. 112543.

Meng, L., X. Qiu, T. Gao, Z. Li, and W. Zhang (2020). "An Inverse Approach to the Accurate Modelling of 3D-printed Sandwich Panels with Lattice Core Using Beams of Variable Cross-Section". In: *Composite Structures* 247, p. 112363.

Meyer, P. P., T. Tancogne-Dejean, and D. Mohr (2024). "Non-Symmetric Plate-Lattices: Recurrent Neural Network-Based Design of Optimal Metamaterials". In: *Acta Materialia* 278, p. 120246.

Mikaeili, E., S. Claus, and P. Kerfriden (2022). "Concurrent Multiscale Analysis without Meshing: Microscale Representation with CutFEM and Micro/Macro Model Blending". In: *Computer Methods in Applied Mechanics and Engineering* 393, p. 114807.

Monaldo, E. and S. Marfia (2021). "Multiscale Technique for the Analysis of 3D-printed Materials". In: *International Journal of Solids and Structures* 232, p. 111173.

Montgomery, S. M., H. Hilborn, C. M. Hamel, X. Kuang, K. N. Long, and H. J. Qi (2023). "The 3D Printing and Modeling of Functionally Graded Kelvin Foams for Controlling Crushing Performance". In: *Extreme Mechanics Letters* 46, p. 101323.

Mora, S., N. M. Pugno, and D. Misseroni (2022). "3D Printed Architected Lattice Structures by Material Jetting". In: *Materials Today* 59, pp. 107–132.

Musenich, L., A. Stagni, and F. Libonati (2023). "Hierarchical Bioinspired Architected Materials and Structures". In: *Extreme Mechanics Letters* 58, p. 101945.

Naceur, H., J. Rahmoun, J. Halgrin, and P. Chabrand (2015). "Multiscale Finite Element Modelling of Ductile Damage Behaviour of the Human Femur under Dynamic Loading". In: *International Journal of Damage Mechanics* 24.3, pp. 418–445.

Nguyen, V. P., C. Anitescu, S. P. Bordas, and T. Rabczuk (2015). "Isogeometric Analysis: An Overview and Computer Implementation Aspects". In: *Mathematics and Computers in Simulation* 117, pp. 89–116.

Nian, Y., S. Wan, P. Zhou, X. Wang, R. Santiago, and M. Li (2021). "Energy Absorption Characteristics of Functionally Graded Polymer-Based Lattice Structures Filled Aluminum Tubes under Transverse Impact Loading". In: *Materials & Design* 209, p. 110011.

Niutta, C. B., R. Ciardiello, F. Berto, D. S. Paolino, and A. Tridello (2022). "On the Influence of Additive Manufacturing Defects on the Energy Absorption Capability of a Lattice Structure". In: *Procedia Structural Integrity*. 23 European Conference on Fracture 42, pp. 1449–1457.

nTop, Inc. (2025). *nTop*. Version 4.0.

Omairey, S. L., P. D. Dunning, and S. Sriramula (2019). "Development of an ABAQUS Plugin Tool for Periodic RVE Homogenisation". In: *Engineering with Computers* 35.2, pp. 567–577.

Opgenoord, M. M. and K. E. Willcox (2018). "Aeroelastic Tailoring Using Additively Manufactured Lattice Structures". In: *2018 Multidisciplinary Analysis and Optimization Conference*. 2018 Multidisciplinary Analysis and Optimization Conference. Atlanta, Georgia: American Institute of Aeronautics and Astronautics.

Orthopedic Implants AM (2024). Today's medical developments. URL: https://www.todaysmedicaldevelopments.com/news/renishaw-additive-manufacturing-orthopedic-implants/.

Ozdemir, M., U. Simsek, G. Kiziltas, C. E. Gayir, A. Celik, and P. Sendur (2023). "A Novel Design Framework for Generating Functionally Graded Multi-Morphology Lattices via Hybrid Optimization and Blending Methods". In: *Additive Manufacturing* 70, p. 103560.

Pagac, M., J. Hajnys, Q.-P. Ma, L. Jancar, J. Jansa, P. Stefek, and J. Mesicek (2021). "A Review of Vat Photopolymerization Technology: Materials, Applications, Challenges, and Future Trends of 3D Printing". In: *Polymers* 13.4, p. 598.

Pan, C., Y. Han, and J. Lu (2020). "Design and Optimization of Lattice Structures: A Review". In: *Applied Sciences* 10.18, p. 6374.

Paszke, A. et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. URL: http://arxiv.org/abs/1912.01703 (visited on 09/16/2025). Pre-published.

Peng, C., P. Tran, and A. P. Mouritz (2022). "Compression and Buckling Analysis of 3D Printed Carbon Fibre-Reinforced Polymer Cellular Composite Structures". In: *Composite Structures* 300, p. 116167.

Phani, A. S. and M. I. Hussein, eds. (2017). *Dynamics of Lattice Materials*. Chichester, West Sussex, United Kingdom: John Wiley & Sons, Inc. 286 pp.

Pilkey, W. D. (2002). *Analysis and Design of Elastic Beams: Computational Methods*. 1st ed. Wiley.

Polak, E. and G. Ribiere (1969). "Note sur la convergence de méthodes de directions conjuguées". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 3.16, pp. 35–43.

Prajapati, M. J., A. Kumar, S.-C. Lin, and J.-Y. Jeng (2023). "Reducing Mechanical Anisotropy in Material Extrusion Process Using Bioinspired Architectured Lattice Structures". In: *Additive Manufacturing*, p. 103480.

PTC, Inc. (2025). *Creo Parametric*. Version 10.0.

Pugalendhi, A., R. Ranganathan, and S. Ganesan (2021). "Impact of Process Parameters on Mechanical Behaviour in Multi-Material Jetting". In: *Materials Today: Proceedings*. International Mechanical Engineering Congress 2019 46, pp. 9139–9144.

Rahman, H., E. Yarali, A. Zolfagharian, A. Serjouei, and M. Bodaghi (2021). "Energy Absorption and Mechanical Performance of Functionally Graded Soft–Hard Lattice Structures". In: *Materials* 14.6 (6), p. 1366.

Raju, K., T. Tay, and V. B. C. Tan (2021). "A Review of the FE2 Method for Composites". In: *Multiscale and Multidisciplinary Modeling, Experiments and Design* 4.

Rao, C. and Y. Liu (2020). "Three-Dimensional Convolutional Neural Network (3D-CNN) for Heterogeneous Material Homogenization". In: *Computational Materials Science* 184, p. 109850.

Riddell Diamond Technology (2019). Riddell. URL: https://content.riddell.com/diamond/.

Riva, L., P. S. Ginestra, and E. Ceretti (2021). "Mechanical Characterization and Properties of Laser-Based Powder Bed–Fused Lattice Structures: A Review". In: *The International Journal of Advanced Manufacturing Technology* 113.3–4, pp. 649–671.

Robert McNeel,
bibinitperiod A. (2025). *Rhinoceros3D*. Version 8.0.

Rodrigo, C., S. Xu, Y. Durandet, D. Fraser, and D. Ruan (2023). "Quasi-Static and Dynamic Compression of Additively Manufactured Functionally Graded Lattices: Experiments and Simulations".

Rosenkranz, M., K. A. Kalina, J. Brummund, W. Sun, and M. Kästner (2024). "Viscoelasticty with Physics-Augmented Neural Networks: Model Formulation and Training

Methods without Prescribed Internal Variables". In: *Computational Mechanics* 74.6, pp. 1279–1301.

Ross, E. and D. Hambleton (2021). "Using Graph Neural Networks to Approximate Mechanical Response on 3D Lattice Structures". In: *Proceedings of AAG2020-Advances in Architectural Geometry* 24, pp. 466–485.

Saadi, M. A. S. R., A. Maguire, N. T. Pottackal, M. S. H. Thakur, M. M. Ikram, A. J. Hart, P. M. Ajayan, and M. M. Rahman (2022). "Direct Ink Writing: A 3D Printing Technology for Diverse Materials". In: *Advanced Materials* 34.28, p. 2108855.

Sadeghi, M., M. Mirdrikvand, G. R. Pesch, W. Dreher, and J. Thöming (2020). "Full-Field Analysis of Gas Flow within Open-Cell Foams: Comparison of Micro-Computed Tomography-Based CFD Simulations with Experimental Magnetic Resonance Flow Mapping Data". In: *Experiments in Fluids* 61.5, p. 124.

Safari, F., A. Kami, and V. Abedini (2022). "3D Printing of Continuous Fiber Reinforced Composites: A Review of the Processing, Pre- and Post-Processing Effects on Mechanical Properties". In: *Polymers and Polymer Composites* 30, p. 09673911221098734.

Sajjad, U., T.-u. Rehman, M. Ali, C. W. Park, and W.-M. Yan (2022). "Manufacturing and Potential Applications of Lattice Structures in Thermal Systems: A Comprehensive Review of Recent Advances". In: *International Journal of Heat and Mass Transfer* 198, p. 123352.

Santos, J., A. Sohouli, and A. Suleman (2024). "Micro- and Macro-Scale Topology Optimization of Multi-Material Functionally Graded Lattice Structures". In: *Journal of Composites Science* 8.4, p. 124.

Savio, G., A. Curtarello, S. Rosso, R. Meneghello, and G. Concheri (2019). "Homogenization Driven Design of Lightweight Structures for Additive Manufacturing". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 13.1, pp. 263–276.

Scalzo, F., G. Totis, E. Vaglio, and M. Sortino (2021). "Experimental Study on the High-Damping Properties of Metallic Lattice Structures Obtained from SLM". In: *Precision Engineering* 71, pp. 63–77.

Schommartz, J. O., D. K. Klein, J. C. A. Cobo, and O. Weeger (2024). *Physics-Augmented Neural Networks for Constitutive Modeling of Hyperelastic Geometrically Exact Beams.* URL: http://arxiv.org/abs/2407.00640. Pre-published.

Schwahofer, O., S. Büttner, D. Colin, and K. Drechsler (2023). "Tailored Elastic Properties of Beam-Based Lattice Unit Structures". In: *International Journal of Mechanics and Materials in Design.*

Scroggs, M. W., I. A. Baratta, C. N. Richardson, and G. N. Wells (2022). "Basix: A Runtime Finite Element Basis Evaluationlibrary". In: *Journal of Open Source Software* 7.73, p. 3982.

Scroggs, M. W., J. S. Dokken, C. N. Richardson, and G. N. Wells (2022). "Construction of Arbitrary Order Finite Element Degree-of-Freedom Maps on Polygonal and Polyhedral Cell Meshes". In: *ACM Transactions on Mathematical Software* 48.2, pp. 1–23.

Seharing, A., A. H. Azman, and S. Abdullah (2020). "A Review on Integration of Lightweight Gradient Lattice Structures in Additive Manufacturing Parts". In: *Advances in Mechanical Engineering* 12.6, p. 1687814020916951.

Seltmann, S. and A. Hasse (2023). "Topology Optimization of Compliant Mechanisms with Distributed Compliance (Hinge-Free Compliant Mechanisms) by Using Stiffness and Adaptive Volume Constraints Instead of Stress Constraints". In: *Mechanism and Machine Theory* 180, p. 105133.

Seo, J. (2024). "Solving Real-World Optimization Tasks Using Physics-Informed Neural Computing". In: *Scientific Reports* 14.1, p. 202.

Shami, T. M., A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili (2022). "Particle Swarm Optimization: A Comprehensive Survey". In: *IEEE Access* 10, pp. 10031–10061.

Shim, P. Y. and S. Manoochehri (1997). "GENERATING OPTIMAL CONFIGURATIONS IN STRUCTURAL DESIGN USING SIMULATED ANNEALING". In: *International Journal for Numerical Methods in Engineering* 40.6, pp. 1053–1069.

Shimoyama, K. and A. Komiya (2022). "Multi-Objective Bayesian Topology Optimization of a Lattice-Structured Heat Sink in Natural Convection". In: *Structural and Multidisciplinary Optimization* 65.1, p. 1.

Siddique, S. H., P. J. Hazell, H. Wang, J. P. Escobedo, and A. A. H. Ameri (2022). "Lessons from Nature: 3D Printed Bio-Inspired Porous Structures for Impact Energy Absorption – A Review". In: *Additive Manufacturing* 58, p. 103051.

Simsek, U., C. E. Gayir, G. Kiziltas, and P. Sendur (2020). "An Integrated Homogenization–Based Topology Optimization via RBF Mapping Strategies for Additively Manufactured FGLS and Its Application to Bandgap Structures". In: *The International Journal of Advanced Manufacturing Technology* 111.5–6, pp. 1361–1374.

Smith, M., Z. Guan, and W. J. Cantwell (2013). "Finite Element Modelling of the Compressive Response of Lattice Structures Manufactured Using the Selective Laser Melting Technique". In: *International Journal of Mechanical Sciences* 67, pp. 28–41.

Somnic, J. and B. W. Jo (2022a). "Homogenization Methods of Lattice Materials". In: *Encyclopedia* 2.2, pp. 1091–1102.

Somnic, J. and B. W. Jo (2022b). "Status and Challenges in Homogenization Methods for Lattice Materials". In: *Materials* 15.2, p. 605.

Song, J., W. Zhou, Y. Wang, R. Fan, Y. Wang, J. Chen, Y. Lu, and L. Li (2019). "Octet-Truss Cellular Materials for Improved Mechanical Properties and Specific Energy Absorption". In: *Materials & Design* 173, p. 107773.

Space Industry AM (2024). Universe Space Tech. URL: https://universemagazine.com/en/additive-manufacturing-how-it-works-in-the-space-industry/.

Specialized and Carbon Saddle (2020). URL: https://www.carbon3d.com/resources/blog/carbon-and-specialized-produce-digitally-printed-saddle.

Stankovic, T., J. Mueller, P. Egan, and K. Shea (2015). "Optimization of Additively Manufactured Multi-Material Lattice Structures Using Generalized Optimality Criteria". In: *Volume 1A: 35th Computers and Information in Engineering Conference*. ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Boston, Massachusetts, USA: American Society of Mechanical Engineers.

Stein, B. van, H. Wang, W. Kowalczyk, M. Emmerich, and T. Bäck (2017). *Cluster-Based Kriging Approximation Algorithms for Complexity Reduction*. URL: http://arxiv.org/abs/1702.01313 (visited on 05/19/2025). Pre-published.

Stollberg, J., T. Gangwar, O. Weeger, and D. Schillinger (2025). "Multiscale Topology Optimization of Functionally Graded Lattice Structures Based on Physics-Augmented Neural Network Material Models". In: *Computer Methods in Applied Mechanics and Engineering* 438, p. 117808.

Sun, B., X. Yan, P. Liu, Y. Xia, and L. Lu (2023). "Parametric Plate Lattices: Modeling and Optimization of Plate Lattices with Superior Mechanical Properties". In: *Additive Manufacturing* 72, p. 103626.

Sun, T., B. Chen, D. Yue, and D. Li (2025). "Design and Strengthening of Dual Bio-Inspired Lattice Structures Based on Additive Manufacturing". In: *Alexandria Engineering Journal* 113, pp. 573–583.

Surjadi, J. U., B. F. G. Aymon, M. Carton, and C. M. Portela (2025). "Double-Network-Inspired Mechanical Metamaterials". In: *Nature Materials*.

Tahmasebimoradi, A., C. Mang, and X. Lorang (2021). "A Numerical Hybrid Finite Element Model for Lattice Structures Using 3d/Beam Elements". In: *ASME 2021 International Mechanical Engineering Congress and Exposition*. Virtuel, France: American Society of Mechanical Engineers.

Tancogne-Dejean, T., M. Diamantopoulou, M. B. Gorji, C. Bonatti, and D. Mohr (2018). "3D Plate-Lattices: An Emerging Class of Low-Density Metamaterial Exhibiting Optimal Isotropic Stiffness". In: *Advanced Materials* 30.45, p. 1803334.

Tannous, M., C. Ghnatios, E. Fonn, T. Kvamsdal, and F. Chinesta (2024). *Machine Learning (Ml) Based Reduced Order Modelling (Rom) for Linear and Non-Linear Solid and Structural Mechanics*. URL: https://www.ssrn.com/abstract=4867758 (visited on 09/04/2025). Pre-published.

Terriault, P. and V. Brailovski (2018). "Modeling and Simulation of Large, Conformal, Porosity-Graded and Lightweight Lattice Structures Made by Additive Manufacturing". In: *Finite Elements in Analysis and Design* 138, pp. 1–11.

Thakolkaran, P., A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, and S. Kumar (2022). "NN-EUCLID: Deep-learning Hyperelasticity without Stress Data". In: *Journal of the Mechanics and Physics of Solids* 169, p. 105076.

Tomita, S., K. Shimanuki, H. Nishigaki, S. Oyama, T. Sasagawa, D. Murai, and K. Umemoto (2023). "Origami-Inspired Metamaterials with Switchable Energy Absorption Based on Bifurcated Motions of a Tachi-Miura Polyhedron". In: *Materials & Design* 225, p. 111497.

Toselli, A. and O. B. Widlund (2005). *Domain Decomposition Methods — Algorithms and Theory*. Vol. 34. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg.

Tyflopoulos, E. and M. Steinert (2020). "Topology and Parametric Optimization-Based Design Processes for Lightweight Structures". In: *Applied Sciences* 10.13, p. 4496.

Vafaeefar, M., K. M. Moerman, and T. J. Vaughan (2025). "LatticeWorks: An Open-Source MATLAB Toolbox for Nonuniform, Gradient and Multi-Morphology Lattice Generation, and Analysis". In: *Materials & Design* 250, p. 113564.

Vaiani, L., A. E. Uva, and A. Boccaccio (2025). "Optimal Lattice Spring Models Derived from Triangular and Tetrahedral Meshes". In: *International Journal of Mechanical Sciences*, p. 110442.

Vaissier, B., J.-P. Pernot, L. Chougrani, and P. Véron (2019). "Genetic-Algorithm Based Framework for Lattice Support Structure Optimization in Additive Manufacturing". In: *Computer-Aided Design* 110, pp. 11–23.

Valizadeh, I. and O. Weeger (2024). "Rate-Dependent Energy Dissipation of Graded Viscoelastic Structures Fabricated by Grayscale Vat Photopolymerization". In: *Smart Materials and Structures* 33.6, p. 065006.

Van De Schoot, R., S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, and C. Yau (2021). "Bayesian Statistics and Modelling". In: *Nature Reviews Methods Primers* 1.1, p. 1.

Van Manen, T., M. Ganjian, K. Modaresifar, L. E. Fratila-Apachitei, and A. A. Zadpoor (2023). "Automated Folding of Origami Lattices: From Nanopatterned Sheets to Stiff Meta-Biomaterials". In: *Small* 19.3, p. 2203603.

Velasco, L., H. Guerrero, and A. Hospitaler (2024). "A Literature Review and Critical Analysis of Metaheuristics Recently Developed". In: *Archives of Computational Methods in Engineering* 31.1, pp. 125–146.

VeroClear (2024). *VeroClear: A Transparent 3D Printing Material*. URL: https://www.stratasys.com/en/materials/materials-catalog/polyjet-materials/veroclear/ (visited on 01/24/2024).

Vijayakumaran, H., J. B. Russ, G. H. Paulino, and M. A. Bessa (2025). "Consistent Machine Learning for Topology Optimization with Microstructure-Dependent Neural

Network Material Models". In: *Journal of the Mechanics and Physics of Solids* 196, p. 106015.

Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3, pp. 261–272.

Vouvakis, M., Z. Cendes, and J.-F. Lee (2006). "A FEM Domain Decomposition Method for Photonic and Electromagnetic Band Gap Structures". In: *IEEE Transactions on Antennas and Propagation* 54.2, pp. 721–733.

Wade, C., D. Beck, and R. MacCurdy (2025). *Implicit Modeling for 3D-printed Multi-material Computational Object Design via Python*. URL: http://arxiv.org/abs/2509.15562 (visited on 10/09/2025). Pre-published.

Wang, H., P. F. M. Tan, Y. Z. Foo, S. Owyong, Z. Lu, H. T. Chen, T. H. Nguyen, and A. Lin (2025). "Integrating Graph Neural Networks with Physics-Informed Loss Function for Mechanical Response Prediction of Hollow Concrete Structures with Morphed Honeycomb Configurations". In: *Materials & Design* 251, p. 113659.

Wang, J. and A. Panesar (2022). "Machine Learning Based Lattice Generation Method Derived from Topology Optimisation". In: *Additive Manufacturing* 60, p. 103238.

Wang, J., J. Zhu, T. Liu, Y. Wang, H. Zhou, and W.-H. Zhang (2023). "Topology Optimization of Gradient Lattice Structure under Harmonic Load Based on Multiscale Finite Element Method". In: *Structural and Multidisciplinary Optimization* 66.9, p. 202.

Wang, J., R. Westermann, X. Gao, and J. Wu (2024). "Design and Optimization of Functionally-Graded Triangular Lattices for Multiple Loading Conditions". In: *Computer Methods in Applied Mechanics and Engineering* 432, p. 117335.

Wang, L., L. He, X. Wang, S. Soleimanian, Y. Yu, G. Chen, J. Li, and M. Chen (2023). "Multiscale Evaluation of Mechanical Properties for Metal-Coated Lattice Structures". In: *Chinese Journal of Mechanical Engineering* 36.1, p. 106.

Wang, Q., J. S. Hesthaven, and D. Ray (2019). "Non-Intrusive Reduced Order Modeling of Unsteady Flows Using Artificial Neural Networks with Application to a Combustion Problem". In: *Journal of Computational Physics* 384, pp. 289–307.

Wang, X., R. Qin, B. Chen, X. Niu, and J. Zhou (2022). "Multi-Scale Collaborative Optimization of Lattice Structures Using Laser Additive Manufacturing". In: *International Journal of Mechanical Sciences* 222, p. 107257.

Wang, X., L. Zhu, L. Sun, and N. Li (2021). "Optimization of Graded Filleted Lattice Structures Subject to Yield and Buckling Constraints". In: *Materials & Design* 206, p. 109746.

Wang, Y., Y. Yu, C. Wang, G. Zhou, A. Karamoozian, and W. Zhao (2020). "On the Out-of-Plane Ballistic Performances of Hexagonal, Reentrant, Square, Triangular and Circular Honeycomb Panels". In: *International Journal of Mechanical Sciences* 173, p. 105402.

Weeger, O., N. Boddeti, S. .-. Yeung, S. Kaijima, and M. L. Dunn (2019). "Digital Design and Nonlinear Simulation for Additive Manufacturing of Soft Lattice Structures". In: *Additive Manufacturing* 25, pp. 39–49.

Weeger, O., I. Valizadeh, Y. Mistry, and D. Bhate (2023). "Inelastic Finite Deformation Beam Modeling, Simulation, and Validation of Additively Manufactured Lattice Structures". In: *Additive Manufacturing Letters* 4, p. 100111.

White, L., X. Liang, G. Zhang, J. Cagan, and Y. J. Zhang (2024). "A Modified Simulated Annealing-Based Method for Hybrid Lattice Support Structure Design in LPBF Additive Manufacturing". In: *Journal of Computing and Information Science in Engineering* 24.12, p. 121002.

White, L., G. Zhang, J. Seo, N. Lamprinakos, A. Rollett, J. Cagan, and Y. J. Zhang (2024). "A Multi-Sized Unit Cell Method for the Design of LPBF Lattice Support Structures Concerning Complex Geometries". In: *Journal of Computing and Information Science in Engineering* 24.12, p. 124502.

Woldseth, R. V., N. Aage, J. A. Bærentzen, and O. Sigmund (2022). "On the Use of Artificial Neural Networks in Topology Optimisation". In: *Structural and Multidisciplinary Optimization* 65.10, p. 294.

Wu, J., O. Sigmund, and J. P. Groen (2021). "Topology Optimization of Multi-Scale Structures: A Review". In: *Structural and Multidisciplinary Optimization* 63.3, pp. 1455–1480.

Wu, J., W. Wang, and X. Gao (2021). "Design and Optimization of Conforming Lattice Structures". In: *IEEE Transactions on Visualization and Computer Graphics* 27.1, pp. 43–56.

Wu, W., D. Qi, W. Hu, L. Xi, L. Sun, B. Liao, F. Berto, G. Qian, and D. Xiao (2020). "Synchrotron X-ray Micro-Computed Tomography Imaging of 3D Re-Entrant Micro Lattice during in Situ Micro Compression Experimental Process". In: *Materials & Design* 192, p. 108743.

Wu, Y., J. Fang, C. Wu, C. Li, G. Sun, and Q. Li (2023). "Additively Manufactured Materials and Structures: A State-of-the-Art Review on Their Mechanical Characteristics and Energy Absorption". In: *International Journal of Mechanical Sciences* 246, p. 108102.

Wu, Y., L. Wang, Z. Li, L. Wu, and Y. Liu (2025). "A Deep Generative Multiscale Topology Optimization Framework Considering Manufacturing Defects and Parametrical Uncertainties". In: *Computer Methods in Applied Mechanics and Engineering* 437, p. 117778.

X VEIN (2020). Autodesk. URL: https://www.autodesk.com/community/gallery/project/154278/x-vein--finally-released-as-open-source-project-.

Xiao, L., W. Song, and X. Xu (2020). "Experimental Study on the Collapse Behavior of Graded Ti-6Al-4V Micro-Lattice Structures Printed by Selective Laser Melting under High Speed Impact". In: *Thin-Walled Structures* 155, p. 106970.

Xu, J., C. Fu, Q. Fu, Y. Chen, Y. Ma, and X. Feng (2021). "Flexible Arc-Armor Inspired by Origami". In: *International Journal of Mechanical Sciences* 201, p. 106463.

Xu, X., Y. Zhang, X. Wang, J. Fang, J. Chen, and J. Li (2022). "Searching Superior Crashworthiness Performance by Constructing Variable Thickness Honeycombs with Biomimetic Cells". In: *International Journal of Mechanical Sciences* 235, p. 107718.

Yang, C., Q. M. Li, and Y. Wang (2021). "Compressive Properties of Cuttlebone-like Lattice (CLL) Materials with Functionally Graded Density". In: *European Journal of Mechanics - A/Solids* 87, p. 104215.

Yang, J., X. Chen, Y. Sun, J. Zhang, C. Feng, Y. Wang, K. Wang, and L. Bai (2022). "Compressive Properties of Bidirectionally Graded Lattice Structures". In: *Materials & Design* 218, p. 110683.

Yang, K., X. Hu, F. Pan, C. Qiao, B. Ding, L. Hu, X. Hu, Z. He, and Y. Chen (2023). "An On-Demand Tunable Energy Absorption System to Resolve Multi-Directional Impacts". In: *International Journal of Solids and Structures* 271–272, p. 112257.

Yang, Z., C. Zhang, W. Liao, T. Liu, and H. Yang (2024). "Efficient and Exquisite Concurrent Optimization of Hierarchical Structures with Non-Uniform Eccentric Body Centered Cubic Lattice". In: *Computer Methods in Applied Mechanics and Engineering* 423, p. 116862.

Yin, H., F. Meng, L. Zhu, and G. Wen (2023). "Optimization Design of a Novel Hybrid Hierarchical Cellular Structure for Crashworthiness". In: *Composite Structures* 303, p. 116335.

Yin, H., W. Zhang, L. Zhu, F. Meng, J. Liu, and G. Wen (2023). "Review on Lattice Structures for Energy Absorption Properties". In: *Composite Structures* 304, p. 116397.

Yousefi, A., S. Jolaiy, M. Lalegani Dezaki, A. Zolfagharian, A. Serjouei, and M. Bodaghi (2022). "3D-Printed Soft and Hard Meta-Structures with Supreme Energy Absorption and Dissipation Capacities in Cyclic Loading Conditions". In: *Advanced Engineering Materials* n/a.n/a, p. 2201189.

Yüksel, N., O. Eren, H. R. Börklü, and H. K. Sezer (2024). "Mechanical Properties of Additively Manufactured Lattice Structures Designed by Deep Learning". In: *Thin-Walled Structures* 196, p. 111475.

Yvonnet, J. and G. Bonnet (2014). "NONLOCAL/COARSE-GRAINING HOMOGENIZATION OF LINEAR ELASTIC MEDIA WITH NON-SEPARATED SCALES USING LEAST-SQUARE POLYNOMIAL FILTERS". In: *International Journal for Multiscale Computational Engineering* 12.5, pp. 375–395.

Zanardi, I., S. Venturi, and M. Panesi (2023). "Adaptive Physics-Informed Neural Operator for Coarse-Grained Non-Equilibrium Flows". In: *Scientific Reports* 13.1, p. 15497.

Zhang, H.-w., J. Wu, and Z. Fu (2010). "Extended Multiscale Finite Element Method for Mechanical Analysis of Periodic Lattice Truss Materials". In: *International Journal for Multiscale Computational Engineering* 8, pp. 597–613.

Zhang, J., Y. Sato, and J. Yanagimoto (2021). "Homogenization-Based Topology Optimization Integrated with Elastically Isotropic Lattices for Additive Manufacturing of Ultralight and Ultrastiff Structures". In: *CIRP Annals* 70.1, pp. 111–114.

Zhang, W. and M. Al Kobaisi (2022). "On the Monotonicity and Positivity of Physics-Informed Neural Networks for Highly Anisotropic Diffusion Equations". In: *Energies* 15.18, p. 6823.

Zhang, Y., M. Li, Z. Qi, R. Chen, Y. Lin, S. Cao, X. Li, R. Tang, and H. Chen (2025). "Nonlinear Mechanics of Horseshoe Microstructure-Based Lattice Design". In: *International Journal of Mechanical Sciences* 285, p. 109781.

Zhang, Y., Z. Zhang, K. Fu, and X. Luo (2022). "Adaptive Defect Detection for 3-D Printed Lattice Structures Based on Improved Faster R-CNN". In: *IEEE Transactions on Instrumentation and Measurement* 71, pp. 1–9.

Zhao, D. and H. Wang (2023). "Topology Optimization of Compliant Mechanisms Considering Manufacturing Uncertainty, Fatigue, and Static Failure Constraints". In: *Processes* 11.10, p. 2914.

Zhao, J., J. Liu, and Y. Liu (2025). "Optimization of Particle Swarm for Force Uniformity of Personalized 3D Printed Insoles". In: *AUTEX Research Journal* 25.1, p. 20250036.

Zhao, M., X. Li, X. Yan, N. Zhou, B. Pang, B. Peng, and Z. Zeng (2025). "Machine Learning Accelerated Design of Lattice Metamaterials for Customizable Energy Absorption". In: *Thin-Walled Structures* 208, p. 112845.

Zhao, M., X. Li, D. Z. Zhang, and W. Zhai (2023). "TPMS-based Interpenetrating Lattice Structures: Design, Mechanical Properties and Multiscale Optimization". In: *International Journal of Mechanical Sciences* 244, p. 108092.

Zhao, M., D. Z. Zhang, Z. Li, T. Zhang, H. Zhou, and Z. Ren (2022). "Design, Mechanical Properties, and Optimization of BCC Lattice Structures with Taper Struts". In: *Composite Structures* 295, p. 115830.

Zheng, L., K. Karapiperis, S. Kumar, and D. M. Kochmann (2023). "Unifying the Design Space and Optimizing Linear and Nonlinear Truss Metamaterials by Generative Modeling". In: *Nature Communications* 14.1 (1), p. 7563.

Zhong, H., H. Mo, Y. Liang, T. Song, C. Li, G. Shen, R. Das, J. Gu, and M. Qian (2024). "Design of Conformal Lattice Metamaterials for Additive Manufacturing". In: *Current Opinion in Solid State and Materials Science* 30, p. 101162.

Zhong, H., R. Das, J. Gu, and M. Qian (2023). "Low-Density, High-Strength Metal Mechanical Metamaterials beyond the Gibson-Ashby Model". In: *Materials Today* 68, pp. 96–107.

Zhong, H., T. Song, C. Li, R. Das, J. Gu, and M. Qian (2023). "The Gibson-Ashby Model for Additively Manufactured Metal Lattice Materials: Its Theoretical Basis, Limitations and New Insights from Remedies". In: *Current Opinion in Solid State and Materials Science* 27.3, p. 101081.

Zhong, K., Z. Wang, J. Cui, X. Yu, M. Zhang, Z. He, Y. Zhao, and J. Zhao (2025). "Mechanical Behavior of Reinforced Al2O3 Lattice Structures: Effects of Structural Parameters from Experiments and Simulations". In: *Thin-Walled Structures* 207, p. 112753.

Zhou, S., K. Zhang, L. Jin, Q. Gao, and W.-H. Liao (2025). "Efficient Data Driven Optimization Framework for Designing B-spline Honeycombs with Excellent Energy Absorption". In: *Thin-Walled Structures* 209, p. 112941.

Zhu, J., H. Zhou, C. Wang, L. Zhou, S. Yuan, and W. Zhang (2021). "A Review of Topology Optimization for Additive Manufacturing: Status and Challenges". In: *Chinese Journal of Aeronautics* 34.1, pp. 91–110.

Appendix A

pyLattice: A design, simulation, and optimization tool for truss lattice structures

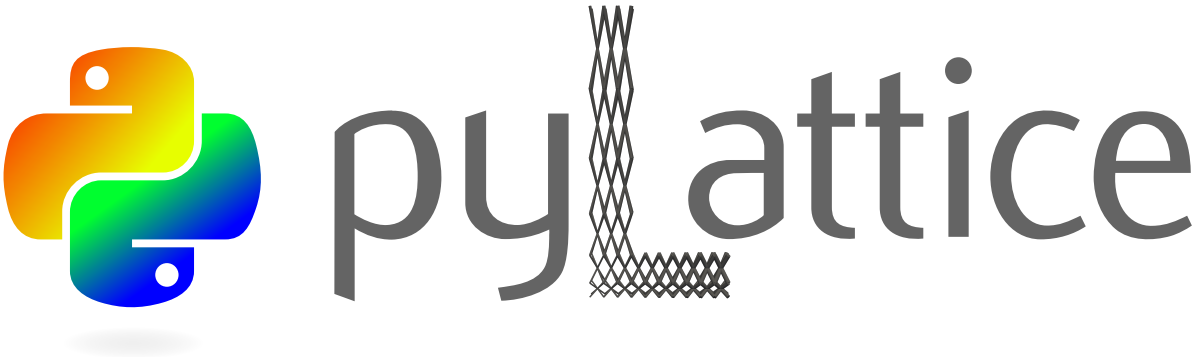

Fig. A.1 pyLattice: A design and simulation tool for truss lattice structures

Throughout this thesis, the development of a dedicated tool for the design, simulation, and optimization of truss lattice structures has proven to be highly valuable. To make this tool available to the community, we have developed a Python package named `pyLattice`. `pyLattice` is a Python™ library that facilitates the generation, analysis, and optimization of truss lattice structures. The package is openly accessible on GitHub at https://github.com/Tcadart/pyLattice, providing the community with a reproducible and extensible framework (see figure A.1).

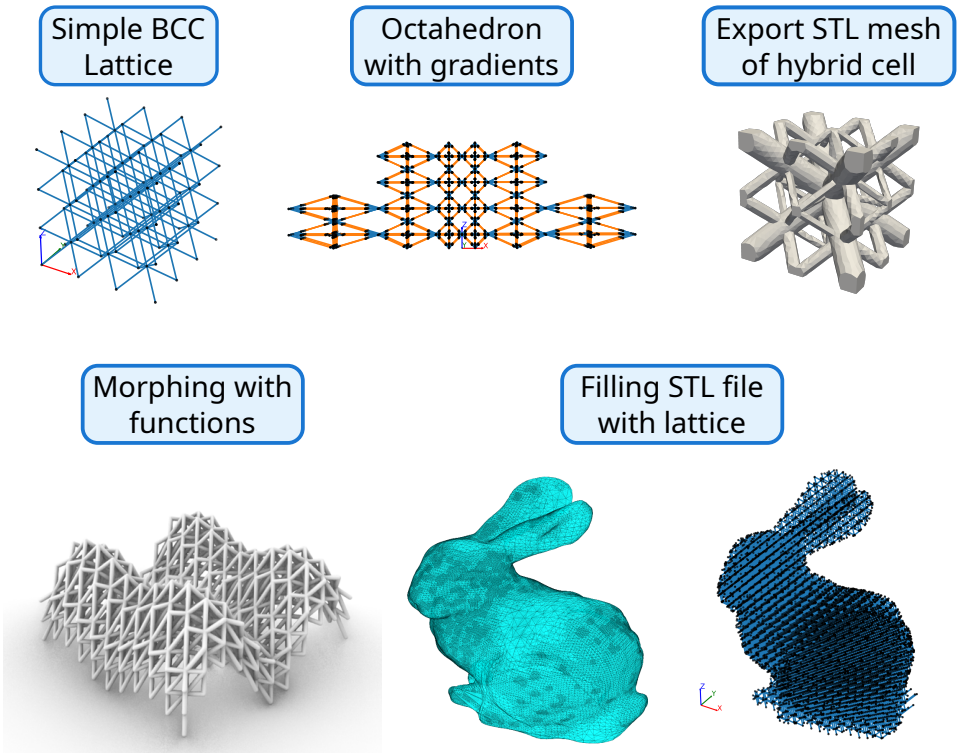

Fig. A.2 Design possibilities with `pyLattice`. The figure illustrates the range of design options available in `pyLattice`, including different lattice geometries, export in `STL` format, morphing onto mathematical functions, and filling a solid geometry with an embedded lattice structure.

A.1 Designing Lattice Structures with pyLattice

The design possibilities of `pyLattice` are extensive, allowing for the creation of various lattice structures. An overview of these different design possibilities is presented in Figure A.2. A lot of different parameters are available to customize the design of the lattice structure:

- **Lattice geometry**: The package supports a wide range of lattice geometries, including BCC, BCCZ, Cubic, Diamond, Octahedron, and more. The user can choose from predefined geometries or create custom ones. Also, the user can create hybrid geometries by combining different base geometries.
- **Beam radius**: The radius of the beams in the structure.
- **Number of cells**: The number of unit cells in each direction of the lattice structure.
- **Cell size**: The size of the unit cell in each direction.

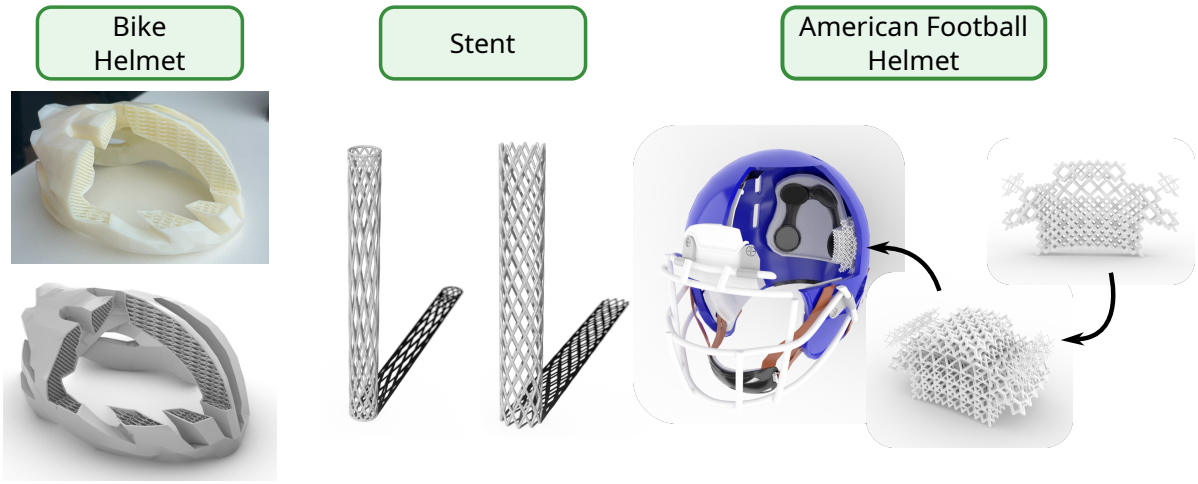

Fig. A.3 Overview of the different applications carried out during this thesis using the `pyLattice` tool.

- **Gradients**: The package allows for the definition of gradients in the beam radius, cell dimensions and material properties. Multiple gradient functions are available, including linear, exponential, and custom functions.
- **Node uncertainty**: The package allows for the definition of a probability distribution for the position of the nodes in the lattice structure. This feature is useful for simulating manufacturing defects or uncertainties in the design.
- **Symmetries**: Definition of symmetries in the lattice structure on multiple planes.
- **Eraser blocks**: The package allows for the definition of eraser blocks, which are used to remove specific cells of the lattice structure. This feature is useful for creating complex geometries.
- **Material properties**: Material properties can be defined for each beam in the lattice structure. The user can choose from predefined materials or create custom ones.

In addition to these design capabilities, the generated lattice structures can be exported in the widely used `STL` format. This format ensures direct compatibility with most slicing software, allowing the structures to be readily prepared for AM. Other advanced transformations of the lattice structure are also implemented. For instance, the lattice can be morphed onto complex shapes defined by mathematical functions. In addition, a lattice structure can be generated inside a solid geometry provided as an `STL`

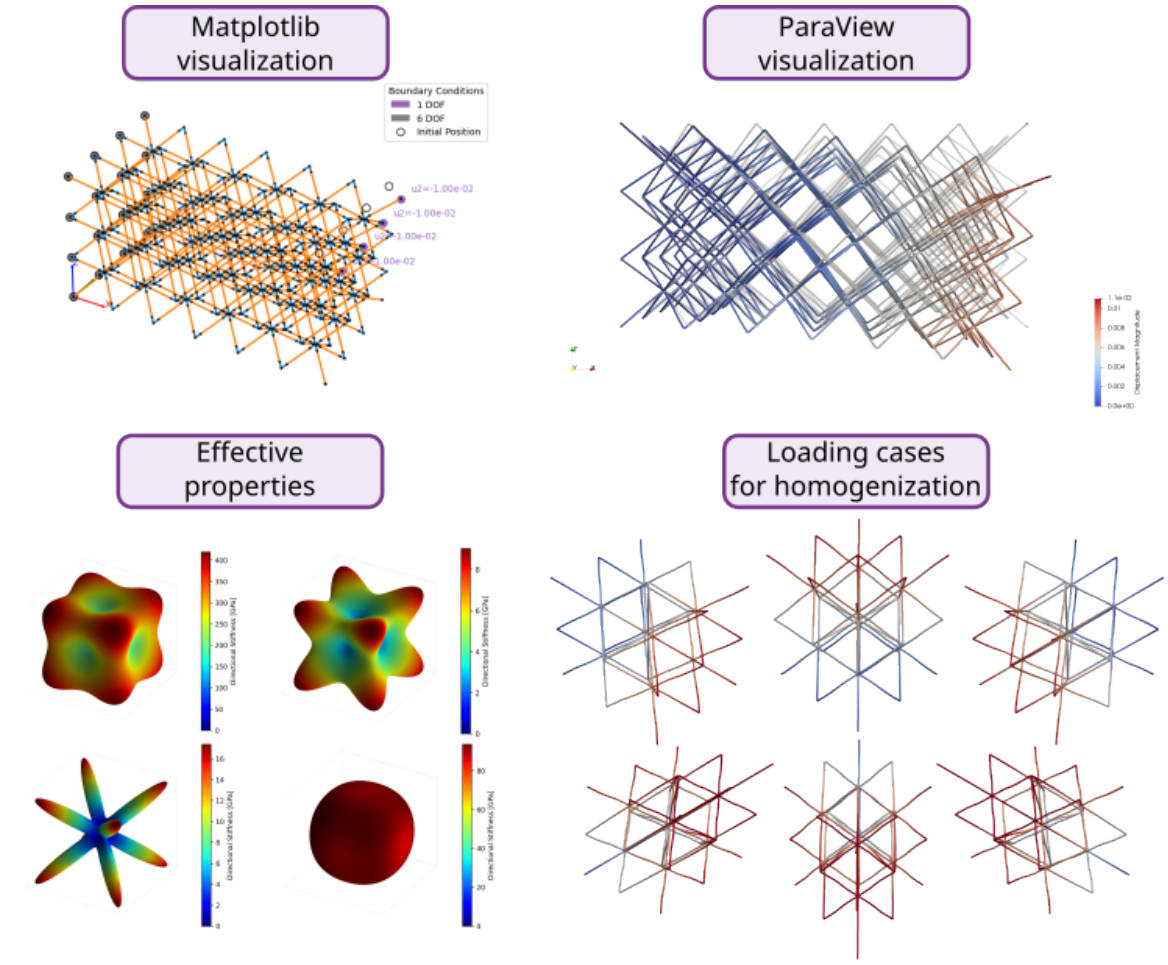

Fig. A.4 Overview of the simulation possibilities available in `pyLattice`, including direct visualization in `matplotlib`, export to ParaView™, computation of effective properties, and elementary loading cases for homogenization.

file, enabling the design of complex components where lattice architectures are embedded within solid domains. To illustrate the versatility of the developed tool, Figure A.3 presents an overview of the different applications carried out during this thesis using `pyLattice`.

A.2 Simulation possibilities with pyLattice

The `pyLattice` package also provides a dedicated simulation module that enables the analysis of lattice structures of various geometries and configurations. This module is built upon the `FEniCSX` library (Alnæs et al. 2014; Baratta et al. 2023; Scroggs, Baratta, et al. 2022; Scroggs, Dokken, et al. 2022), offering a flexible and efficient framework

for FE simulations. The beam penalization method described in Chapter 3 has been implemented within the package. This approach improves the accuracy of lattice structure simulations by accounting for the reduced stiffness of joints, a key aspect in capturing their mechanical response more realistically. Multiple simulation strategies are available in the package to analyze lattice structures:

- **FEM on full-scale lattice structures**: The package enables the simulation of complete lattice structures using the FEM approach.
- **Homogenization**: A homogenization module is provided to compute the effective mechanical properties of unit cells. The use of these homogenized properties for structural simulations is not yet available, but will be implemented in future developments.
- **Schur complement method**: The package includes functionalities for computing the Schur complement, which is particularly relevant for domain decomposition methods.
- **Domain decomposition**: Although the DDM module is not yet implemented, its integration is planned for upcoming releases.

Simulation results can be directly visualized within the package using `matplotlib`, or exported to ParaView™ for more advanced visualization capabilities. An overview of the simulation possibilities is presented in Figure A.4.

A.3 Optimization possibilities with pyLattice

The optimization capabilities of this thesis are implemented within the `pyLattice` package (See figure A.5). This module provides a flexible framework for the design optimization of lattice structures with respect to specific performance criteria. It supports a variety of optimization objectives, constraints, and algorithms, offering users a wide range of possibilities to adapt the workflow to their needs. Several strategies for defining the optimization variables are implemented and are fully compatible with the hybrid geometry cell method introduced earlier:

- Optimization with a constant beam radius,
- Optimization based on gradient functions, such as linear, polynomial, or sinusoidal variations,

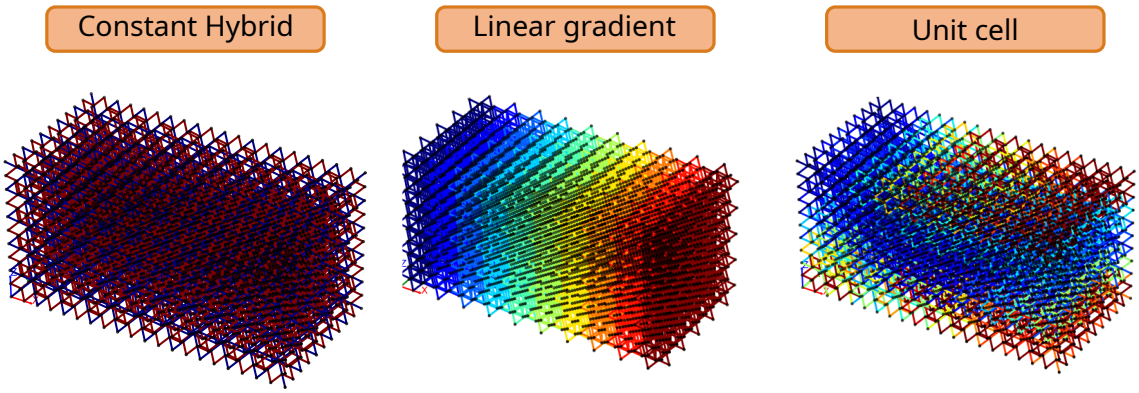

Fig. A.5 Overview of the optimization possibilities available in `pyLattice`.

- Optimization at the unit cell level, where each cell can be independently parameterized.

Furthermore, the simulation strategies described in Section A.2 are directly integrated within the optimization module, enabling consistent evaluation of lattice performances during the optimization loop. The overall framework has been designed to be modular and extensible, allowing the integration of additional optimization algorithms or problem formulations without altering the core structure of the code.

A.4 Pyrough compatibility

`Pyrough` (Iteney et al. 2024) is an open-source Python package that generates self-affine surface roughness on three-dimensional geometries using the classical roughness theory (See figure A.6). In the context of this work, `Pyrough` complements `pyLattice` by directly importing meshes produced with `pyLattice` and enriching their surfaces with a controlled rough topography. The roughness can be prescribed through statistically meaningful parameters, such as the Hurst exponent and the root-mean-square height, which enables systematic studies of the sensitivity of lattice structures to non-ideal surface morphologies. This coupling between `pyLattice` and `Pyrough` allows for the study of the impact of surface roughness on the mechanical behavior of lattice structures, particularly in the context of additive manufacturing. This collaboration was carried out directly with the project leader, Dr. Jonathan Amodeo, CNRS researcher at Aix-Marseille University.

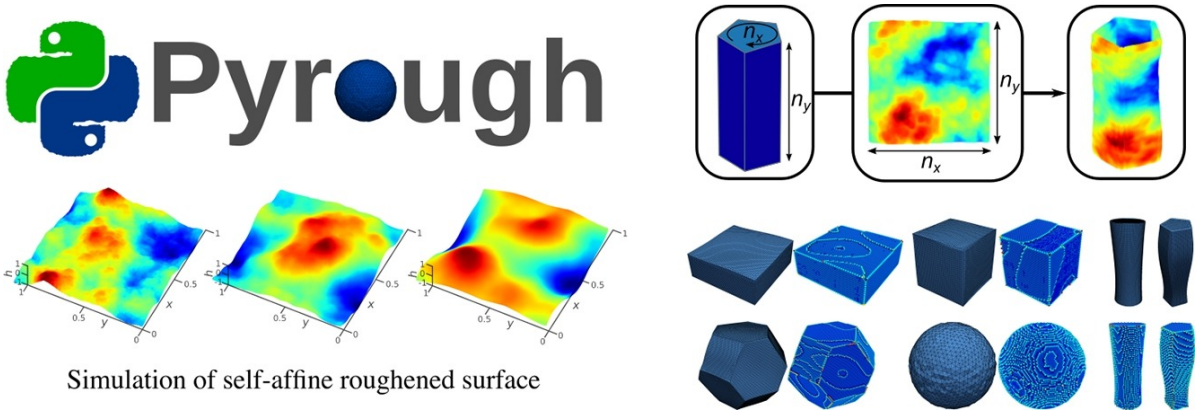

Fig. A.6 Overview of the Pyrough package capabilities. Reproduced with permission from (Iteney et al. 2024).

Appendix B

Analytical Derivation of Radial Basis Function Derivatives with Thin-Plate Spline Kernel

We consider the interpolant

$$f(\boldsymbol{x}) \;=\; \sum_{i=1}^{N} w_i\,\phi\Big(r_i(\boldsymbol{x})\Big) \;+\; c_0 \;+\; \boldsymbol{c}^{\top}\boldsymbol{x}, \qquad r_i(\boldsymbol{x}) \;=\; \|\boldsymbol{x}-\boldsymbol{x}_i\|_2, \tag{B.1}$$

where $\boldsymbol{x}, \boldsymbol{x}_i \in \mathbb{R}^d$, $w_i \in \mathbb{R}$, and $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a radial kernel. The gradient $\nabla f(\boldsymbol{x})$ follows from the chain rule applied to each radial term together with the gradient of the Euclidean norm.

Step 1: gradient of the distance r_i.

Write $r_i(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{x}_i)^{\top}(\boldsymbol{x}-\boldsymbol{x}_i)}$. For $r_i > 0$,

$$\nabla r_i(\boldsymbol{x}) \;=\; \nabla\Big[\,(\,(\boldsymbol{x}-\boldsymbol{x}_i)^{\top}(\boldsymbol{x}-\boldsymbol{x}_i)\,)^{1/2}\Big] \;=\; \frac{1}{2}\,(\cdot)^{-1/2}\,\nabla\Big[(\boldsymbol{x}-\boldsymbol{x}_i)^{\top}(\boldsymbol{x}-\boldsymbol{x}_i)\Big]. \tag{B.2}$$

Since $\nabla\Big[(\boldsymbol{x}-\boldsymbol{x}_i)^{\top}(\boldsymbol{x}-\boldsymbol{x}_i)\Big] = 2(\boldsymbol{x}-\boldsymbol{x}_i)$, we get

$$\nabla r_i(\boldsymbol{x}) \;=\; \frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sqrt{(\boldsymbol{x}-\boldsymbol{x}_i)^{\top}(\boldsymbol{x}-\boldsymbol{x}_i)}} \;=\; \frac{\boldsymbol{x}-\boldsymbol{x}_i}{r_i(\boldsymbol{x})}, \qquad r_i > 0. \tag{B.3}$$

At $\boldsymbol{x} = \boldsymbol{x}_i$, one has $r_i = 0$ and the Euclidean norm is not differentiable, however the limit $\lim_{\boldsymbol{x}\to\boldsymbol{x}_i} \nabla r_i(\boldsymbol{x})$ exists in the sense that every directional derivative is bounded and the contribution of the ith term to ∇f vanishes for the thin plate spline kernel, see below.

Step 2: chain rule for a radial term. For each i,

$$\nabla\Big[\phi\Big(r_i(\boldsymbol{x})\Big)\Big] \;=\; \phi'\Big(r_i(\boldsymbol{x})\Big)\,\nabla r_i(\boldsymbol{x}) \;=\; \phi'(r_i)\,\frac{\boldsymbol{x}-\boldsymbol{x}_i}{r_i}, \qquad r_i>0. \tag{B.4}$$

Step 3: gradient of the full interpolant. Summing (B.4) over i and adding the polynomial tail,

$$\nabla f(\boldsymbol{x}) \;=\; \sum_{i=1}^{N} w_i\,\phi'(r_i)\,\frac{\boldsymbol{x}-\boldsymbol{x}_i}{r_i} \;+\; \boldsymbol{c}, \qquad r_i=\|\boldsymbol{x}-\boldsymbol{x}_i\|_2. \tag{B.5}$$

Specialization to the thin plate spline kernel. For the polyharmonic thin plate spline,

$$\phi(r)=r^2\log r, \qquad \phi'(r)=2r\log r+r, \qquad \frac{\phi'(r)}{r}=2\log r+1, \quad r>0. \tag{B.6}$$

Substituting into (B.5) yields the numerically stable form

$$\nabla f(\boldsymbol{x}) \;=\; \sum_{i=1}^{N} w_i\,\Big(2\log r_i+1\Big)\,(\boldsymbol{x}-\boldsymbol{x}_i) \;+\; \boldsymbol{c}. \tag{B.7}$$

At a center $\boldsymbol{x}=\boldsymbol{x}_i$, the factor $(2\log r_i+1)(\boldsymbol{x}-\boldsymbol{x}_i)$ tends to $\mathbf{0}$ as $r_i\to 0$, hence the contribution of the ith term vanishes in the limit and the gradient remains well defined by continuity.

Titre : Chaîne numérique couplant décomposition de domaine et modèle de substitution pour l'optimisation localisée de structures lattices.

Mots clés : Structures lattice, Structures treillis, Chaîne numérique, Modèle poutre, Optimisation, Modèle de substitution

Résumé :

Les progrès récents des technologies de fabrication, et en particulier de la fabrication additive, ont ouvert de nouvelles perspectives pour la conception de structures complexes. Parmi elles, les structures lattices ont suscité un intérêt considérable, car elles permettent la création de structures périodiques présentant des propriétés et des fonctionnalités auparavant non réalisable. Cependant, la conception et l'optimisation de telles géométries complexes afin d'obtenir des comportements spécifiques demeurent particulièrement difficiles, principalement en raison du coût de calcul élevé inhérent aux chaînes numériques classiques. Pour analyser et comprendre efficacement les structures lattices, de nouvelles chaînes de calcul doivent donc être développés. Le premier chapitre de cette thèse s'est concentré sur la simulation de structures lattices de type treillis modélisées à l'aide d'éléments poutres. La principale difficulté de cette approche réside dans la représentation précise de la raideur des jonctions, un aspect particulièrement complexe à modéliser avec des éléments unidimensionnels. Une méthode généralisée d'ajustement de la raideur des jonctions a été développée en définissant une zone de pénalisation dépendante du rayon des poutres et des angles entre celles-ci. Cette stratégie améliore significativement la cohérence entre les simulations numériques et les résultats expérimentaux. Sur cette base, une approche améliorée couplant la décomposition de domaine et les modèles de substitution a été développée afin d'accroître l'efficacité numérique. Les structures lattices se prêtent naturellement à une décomposition cellule par cellule, ce qui permet à la méthode de décomposition de domaine de réduire considérablement le coût de calcul par rapport aux approches classiques. De plus, le calcul préliminaire du comportement mécanique de cellules unitaires paramétrées en fonction de leur géométrie, combiné à l'utilisation de modèles de substitution, permet de prédire efficacement la réponse mécanique globale de la structure. Deux stratégies distinctes de modèle de substitution ont été introduites : l'une fondée sur la réduction de modèle, l'autre sur les réseaux de neurones enrichis par la physique, chacune apprenant de manière différente la relation entre les paramètres géométriques et les réponses mécaniques. Cette approche se prête particulièrement bien aux cadres d'optimisation pour deux raisons principales : elle est numériquement efficace et, grâce au modèle de substitution, la sensibilité des paramètres de conception peut être directement obtenue. Enfin, un cadre d'optimisation intégrant l'ensemble des méthodes présentées a été développé et testé sur plusieurs cas de référence. L'algorithme d'optimisation basé sur le gradient a démontré sa robustesse, sa précision et son efficacité pratique, tant pour des problèmes de conception simples que de haute dimension. Dans l'ensemble, cette thèse présente une chaîne numérique unifiée et efficace, combinant une approche avancée de simulation à la flexibilité de conception offerte par les structures lattices.

Title: Surrogate-enhanced Domain Decomposition Framework for Local Optimization of Lattice Structures

Keywords: Lattice structures, Numerical methods, Beam models, Optimization, Domain decomposition, Surrogate models

Abstract:

Advances in manufacturing technologies, and in particular additive manufacturing, have opened new possibilities for designing highly complex structures. Among them, lattice structures have gained considerable attention as they enable the creation of material architectures exhibiting properties and functionalities that were previously unattainable. However, designing and optimizing such complex shapes to achieve specific behaviors remains highly challenging, mainly due to the significant computational burden inherent to classical numerical frameworks. To efficiently analyze and understand lattice structures, new computational frameworks must therefore be developed. The first chapter of this thesis focused on the simulation of truss-based lattice structures modeled using beam elements. The main difficulty associated with this approach lies in the accurate representation of joint stiffness, which is particularly challenging when using one-dimensional elements. A generalized joint stiffening method was developed by defining a penalization zone based on the beam radius and the angle between connected beams. This strategy significantly improves the consistency between numerical simulations and experimental results. Building on this work, an enhanced approach that couples domain decomposition with surrogate modeling techniques was developed to further improve computational efficiency and scalability. Lattice structures are naturally decomposed on a cell-by-cell basis, which allows the domain decomposition method to significantly reduce the computational burden compared to classical approaches. Moreover, precomputing the mechanical behavior of shape-parameterized unit cells and coupling them with surrogate models enables the efficient prediction of the global mechanical response. Two distinct surrogate modeling strategies were introduced: one based on reduced-order modeling and the other on physics-augmented neural networks, each capturing the relationship between geometric parameters and mechanical responses in a different manner. This approach is particularly well suited to optimization frameworks for two main reasons. It is computationally efficient, and thanks to the surrogate model, the sensitivity of the design parameters can be directly recovered. Finally, an optimization framework integrating all the previously presented methods was developed and tested on multiple benchmark problems. The gradient-based optimization algorithm demonstrated robustness, accuracy, and practical efficiency in both simple and high-dimensional design spaces. Experimental tests were conducted as a final step to assess the performance of the proposed framework. Overall, this thesis presents a unified and computationally efficient framework that couples an advanced simulation approach with the design flexibility of lattice structures.