Proceedings of the 11th World Congress on Mechanical, Chemical, and Material Engineering (MCM'25)
Paris, France - August, 2025
Paper No. ICMIE 126
DOI: 10.11159/icmie25.126

Rhinoceros-Based Tool for Versatile Lattice Structure Generation

Rahali Yosra[1], Gely Benoît[1]
[1]Capgemini Engineering
4 Avenue Didier Daurat, Blagnac, France
Yosra.rahali@capgemini.com; Benoit.gely@capgemini.com

Abstract - Industries across various sectors, such as aerospace and automotive, are striving to reduce their environmental impact through innovative design and material solutions. In this context, additive manufacturing has enabled the development of lattice structures—architected parts offering high performance and low mass. However, challenges remain in generating diverse lattice designs that accommodate both truss and surface structures, especially when using multiple materials.

This study introduces two complementary methods for lattice generation. The first is a graphical tool called "Next Lattice Design," developed with Grasshopper and the Human UI module. This intuitive interface enables users to generate unit cells and simple specimens, such as cubic or cylindrical shapes, by adjusting design parameters like thickness, wavelength, and discretization fineness. The tool supports STL and INP file exports, facilitating both 3D printing and finite element method (FEM) simulations. The second method leverages Python scripts integrated with Grasshopper to automate lattice generation and control multi-material aspects, supporting the creation of large, diverse databases for machine learning applications.

These methods address limitations in existing solutions by enabling both surface and truss lattice generation while managing multi-material designs. They also provide a detailed visualization of the generated structures, including parameters like relative density and connection integrity. Together, these tools represent a significant advancement in lattice design, paving the way for the integration of complex geometries and diverse material combinations.

***Keywords*:** Lattice structure, Additive manufacturing, Python scripting, Multi-material design.

1. Introduction

Over the past two decades, the rapid advancements in additive manufacturing (AM) technologies have garnered significant attention due to their transformative capabilities. These technologies offer numerous advantages, including the ability to produce near-net-shaped parts with minimal post-processing, thereby reducing material waste and enhancing efficiency. Additionally, AM provides unparalleled design freedom, enabling the fabrication of intricate geometries, such as lattice structures, that are challenging or impossible to achieve using conventional manufacturing techniques. This flexibility has paved the way for innovations like topology optimization, lightweighting, and part consolidation [1,2,3,4].

Characterized by a periodic arrangement of material and void spaces, lattice structures exhibit distinctive physical properties that cannot be achieved with bulk materials. For example, many of these structures demonstrate exceptional strength-to-weight ratios or thermal management capabilities, making them ideal for various applications, including mechanical systems, thermal management solutions, and acoustic devices [5,6]. Additionally, the integration of lattice structures into designs challenges the boundaries of traditional material property charts, such as the Ashby diagram, potentially creating new clusters of materials with optimized performance [7].

Lattice structures can be broadly categorized into two main families: strut-based lattices and Triply Periodic Minimal Surface (TPMS) lattices. Strut-based lattices rely on networks of bars or struts to form their structural framework, offering versatility and ease of design. In contrast, TPMS lattices are defined by their smooth, periodic surfaces that minimize material use while maximizing strength and other functional properties. Both types are widely studied and applied, yet they cater to different design and performance requirements depending on the application. Examples of TPMS lattice structures are illustrated in Figure 1.

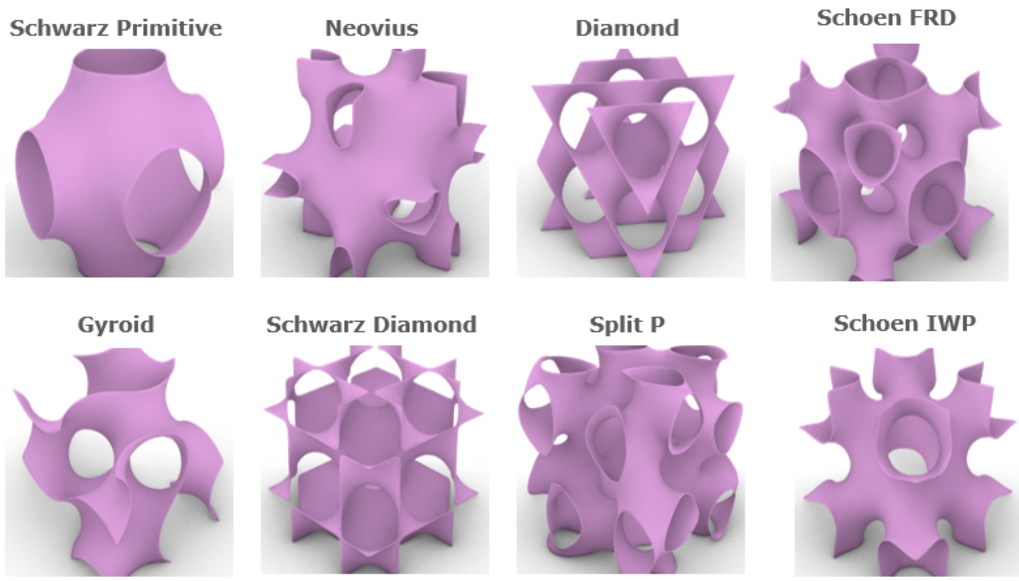

Fig. 1: Examples of Triply Periodic Minimal Surface (TPMS) lattice structures.

Several lattice generation methods have been proposed in the literature. While CAD tools like CATIA and SolidWorks are often employed for generating parametric lattice structures, they are primarily suited for studying a limited number of elementary lattices and producing basic parametric files. These tools often lack the scalability and flexibility required for generating complex lattice geometries, resulting in time-consuming and manual processes that hinder topological optimization [8-11].

Recent advancements have introduced more specialized solutions. For example, FreeCAD plugins support the generation of simple shapes like cylinders, spheres, and cuboids. However, they are limited by their inability to combine multiple lattice shapes or create finite element (FE) meshes, restricting their applicability in advanced engineering applications [12]. MSLattice, a Matlab-based tool, facilitates the design of hybrid and graded lattice while allowing customization of geometrical parameters. However, similar to other tools, MSLattice relies on external commercial software to integrate TPMS structures into arbitrary CAD models, presenting a significant limitation for complex applications such as heat exchanger designs [13].

Other tools like ASLI (A Simple Lattice Infiller), FlattPack, and RegionTPMS have further expanded the possibilities for lattice generation. ASLI enables the creation of lattice infills for arbitrary geometries, while FlattPack offers STL exports and computational meshes for finite element analysis. However, FlattPack's closed-source nature and ASLI's focus on infill lattices rather than structural integration present constraints for broader usage [14, 15]. RegionTPMS, which operates through Mathematica notebooks, supports the generation of multiphase and graded scaffolds with adjustable resolution, but it relies on access to commercial Mathematica software, limiting its accessibility [16].

Advanced commercial tools like Ntopology excel in creating complex lattice designs with features like thickness gradients, seamless integration with FE software, and optimized outputs tailored for additive manufacturing. However, despite its capabilities, Ntopology's high subscription cost and steep learning curve limit its accessibility, particularly for small businesses, individual researchers and academic institutions.

To address these gaps, our work focuses on leveraging Grasshopper, a parametric design module within Rhinoceros. Grasshopper allows for algorithmic modeling of complex shapes, offering significant design flexibility and customization. By developing a tailored lattice generation tool within this framework, our approach overcomes the limitations of existing solutions. It combines user accessibility with the ability to generate diverse lattice structures, optimize material properties, and ensure compatibility with 3D printing and FE analyses, ultimately paving the way for advanced applications in structural and functional design.

2. Method

To address the challenges in lattice structure generation and meet specific design requirements, two complementary solutions have been developed. These solutions are designed to provide both flexibility and usability, ensuring compatibility with fabrication and analysis processes.

The first solution is a graphical user interface developed in Grasshopper, which simplifies lattice design for novice users while maintaining advanced customization capabilities. The second is a Python-GH workflow designed to automate the process of generating large datasets of lattice structures, which are essential for training machine learning models. This workflow enables the creation of varied lattice designs with different parameters, ensuring that a diverse and representative dataset is available for model training and optimization. Together, these methods address the limitations of existing tools and create a robust framework for lattice design.

2.1. Lattice Sandbox Explorator

To facilitate the generation of lattices, a graphical tool called "*Next Lattice Design*" has been developed using Grasshopper's Human UI module. This tool provides an intuitive and user-friendly interface for designing lattice structures with a wide range of customizable options. Figure 2 displays the developed tool, which can be divided into four main parts:

a. Unit Lattice Selection: Users can choose the type of lattice to generate, whether it is a surface-based (e.g., TPMS) or truss-based lattice, as well as generate cylindrical specimens for specific applications.

b. Design Parameters: This section allows users to define critical parameters such as wavelength, parameter *t* which controls surface expansion, thickness, and discretization fineness for export.

c. Export Options: The interface supports exporting the generated lattice in:

- STL format for 3D printing, ensuring compatibility with standard additive manufacturing workflows.
- INP format for Abaqus, enabling finite element analysis.

Users can choose to export TPMS lattices as 2D elements (triangular S3 or quadrilateral S4) and truss lattices as 1D elements with options for square or circular cross-sections.

d. Visualization and Metrics: The final section of the interface provides a real-time preview of the generated lattice, along with two key indicators:

- Relative Density: Displays the lattice's volume fraction, offering insights into its structural density.
- Connectivity Indicator: Alerts users to any unconnected lattice fragments that may compromise the design's functionality or manufacturability.

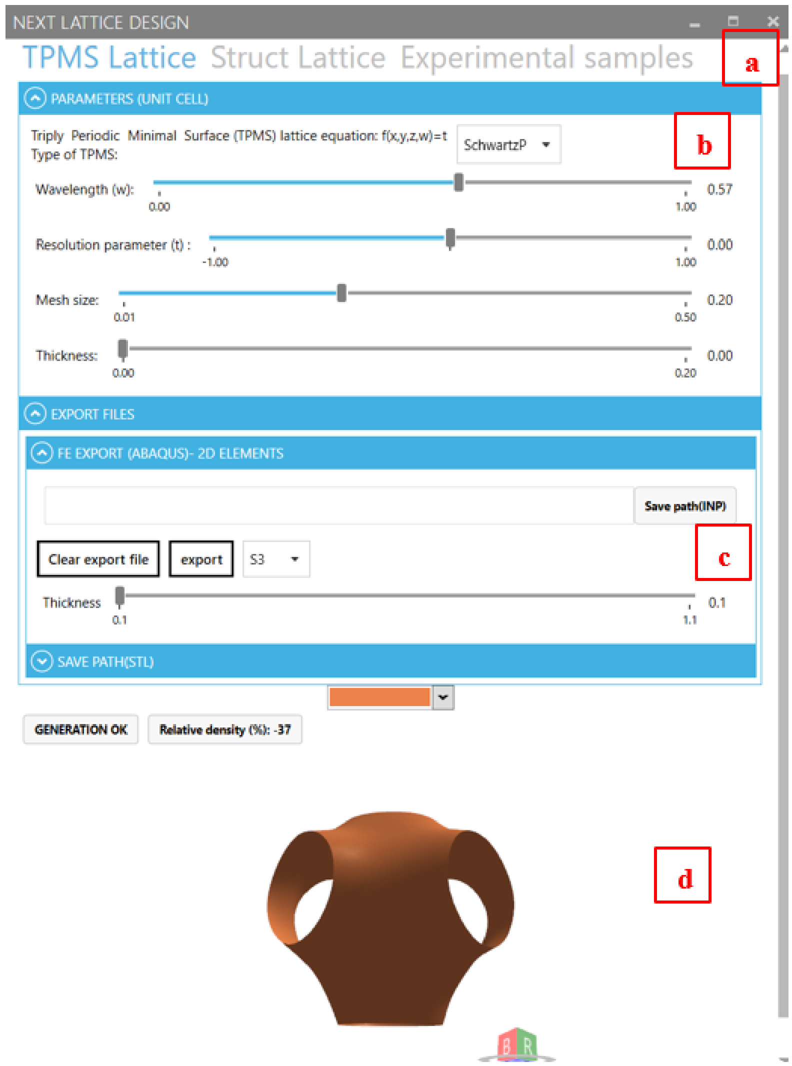

Fig. 2: GUI for the generation of lattice structures.

This interface enables users to define parameters and create diverse lattice structures for various applications. For instance, cylindrical samples designed with the GUI (Figure 3) have been used to characterize the acoustic absorption properties of gyroid-shaped porous structures.

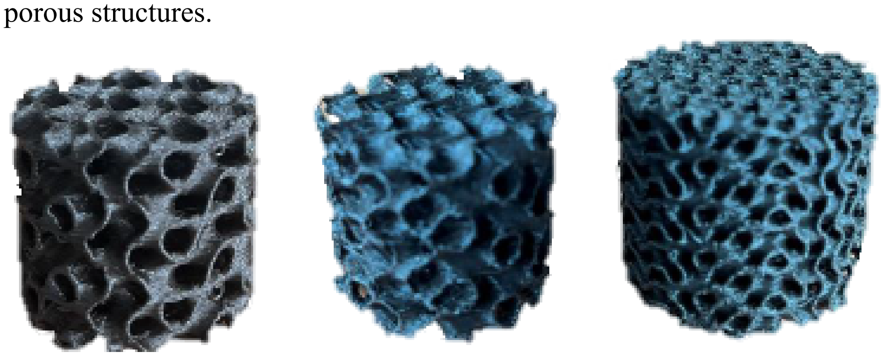

Fig. 3: Acoustic sample designs created with the GUI.

During preliminary 3D printing experiments, it was observed that the slicer introduced erratic movements, resulting in lower print quality compared to solid parts. To address this, a proposed solution involves directly exporting G-code from Grasshopper. Although this approach has been numerically tested on a single sample, further experimental validation is required.

In the context of applications at Capgemini, only the TPMS lattice generation functionality has been validated thus far. Exploring truss lattice generation within the GUI would help identify areas for improvement and expand its capabilities.

Additionally, the Relative Density indicator offers a valuable metric for understanding lattice volume. Comparing these volume-based results to mass-based measurements obtained from 3D-printed unit structures could provide further insights into lattice performance and optimization.

2.2. Python-GH workflow: Design & Multi-Material Control

The graphical interface described earlier enables the creation of individual lattices with user-defined parameters. However, for applications such as machine learning, the ability to generate large datasets is crucial, as these models require extensive data to train effectively and capture the complex patterns inherent in the lattice structures. To meet this need, a Python-Grasshopper (Python-GH) workflow has been developed. This workflow facilitates not only lattice generation but also the introduction of multi-material capabilities.

The Python-GH workflow begins with Python scripts generating random lattice parameters, which are then transmitted to Grasshopper for execution. A failed attempt occurs when the generated parameters result in a lattice structure that is not fully connected, such as multiple disconnected lattice pieces. These disconnected lattices do not meet the desired structural criteria and are considered invalid for further use. This process creates databases of both successful and failed attempts, which are crucial for refining the parameter generation algorithm. To avoid redundant failures, the script checks if a set of parameters has previously led to a disconnected lattice before sending them to Grasshopper. Figure 4 illustrates the initial stage of dataset generation, focusing on the lattice design process.

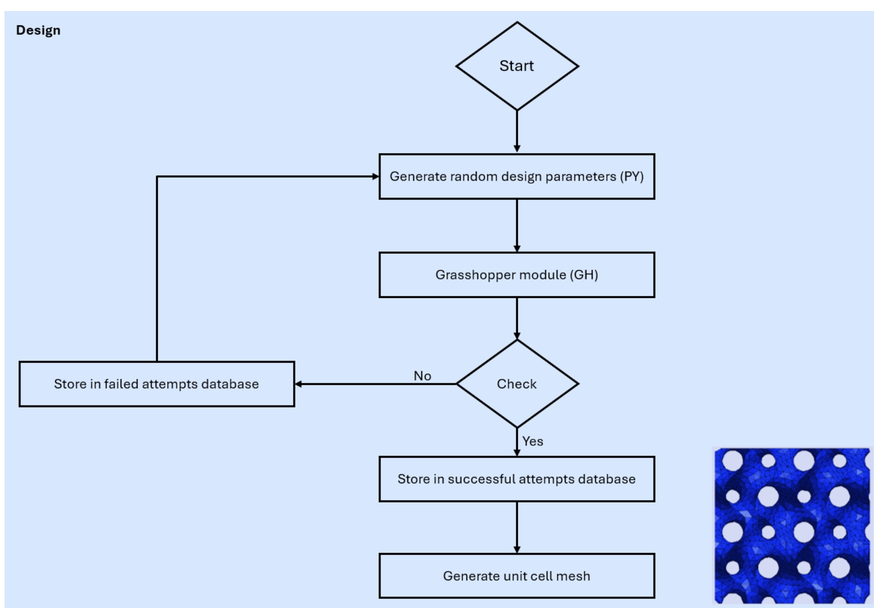

Fig. 4: Diagram illustrating the Python-Grasshopper procedure for lattice design.

To evaluate the functionality of this workflow, experiments were conducted on various lattice types, including Gyroid, Diamond, and Primitive shapes. For Gyroid lattices, as shown in Figure 5(a), initial generation attempts resulted in a success/failure ratio of 0.18 after 220 attempts. A subsequent round of 135 attempts improved the ratio to 0.27, demonstrating the iterative refinement's effectiveness. Similar trends were observed for Diamond lattices, with an initial success/failure ratio of 0.13 over 200 attempts (Figure 5(b)). Primitive lattices demonstrated the highest success rates, achieving a success/failure ratio of 0.71 after 215 attempts (Figure 5(c)), highlighting better compatibility with the explored parameter ranges.

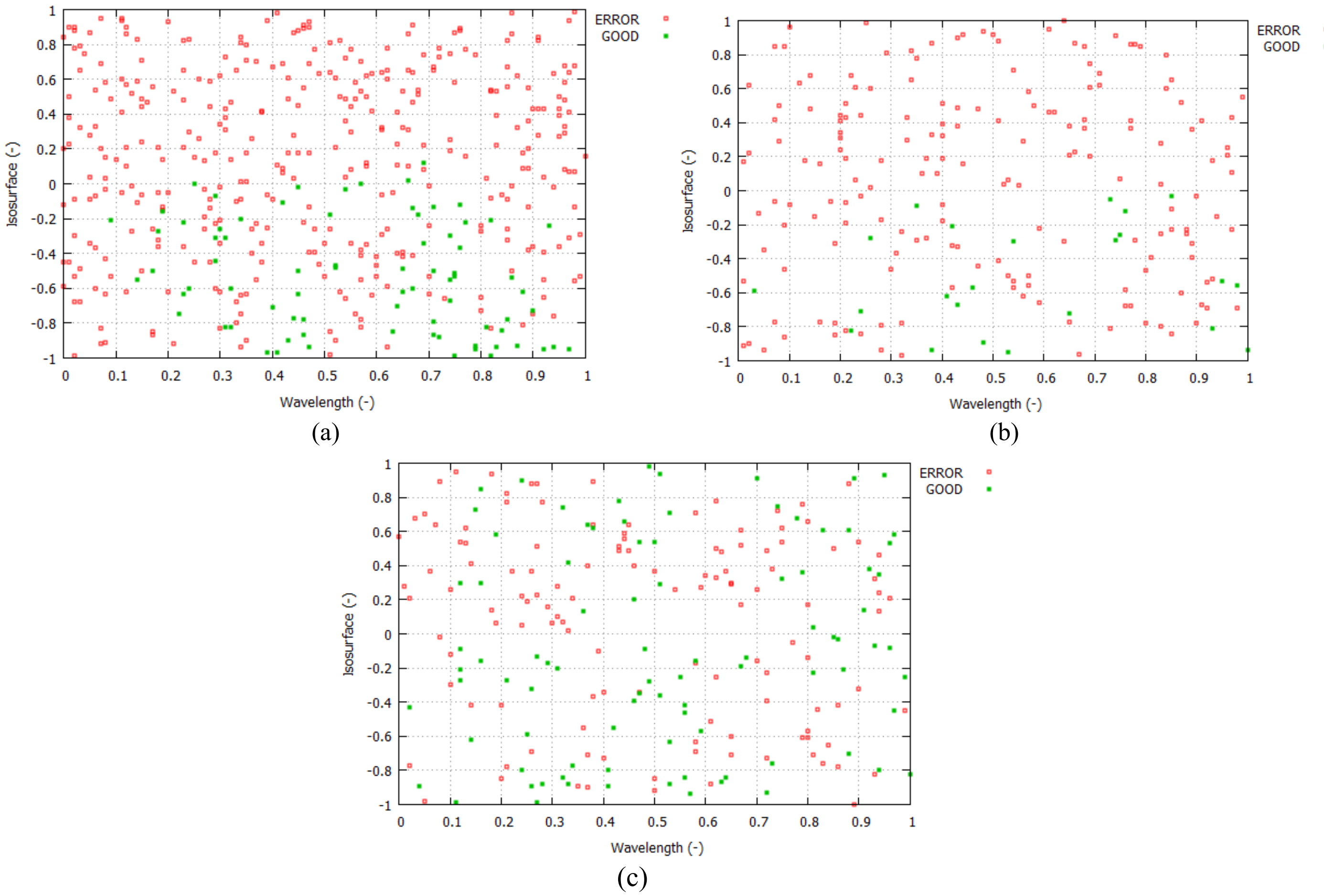

Fig. 5: Summary of successful and failed generation attempts with Python-Grasshopper for (a) Gyroid lattices, (b) Diamond lattices, and (c) Primitive lattices.

The results revealed critical parameter thresholds. For example, with the Gyroid lattice, values of the parameter *t*, which controls surface expansion, exceeding 0.2 frequently resulted in generation failures. To refine these thresholds, further systematic attempts can be conducted by exploring parameter ranges in smaller increments and logging outcomes to identify boundaries where success rates improve. This ensures a more comprehensive understanding of the parameter space and its influence on lattice quality.

The mesh density of the generated lattices depends on the wavelength parameter. Smaller wavelengths produce more intricate geometries, requiring finer meshes for accurate representation. To address this, the mesh size is automatically adjusted based on the wavelength, ensuring a balance between geometric detail and computational efficiency.

The workflow also extends to introducing multi-material capabilities. A Python script was developed to facilitate material distribution, as illustrated in Figure 6. This methodology includes three main approaches: uniform distribution, Gaussian distribution, and coordinate-based distribution. The uniform distribution splits the lattice into two identical material sets (50-50 distribution). The Gaussian distribution introduces a gradual material transition based on statistical parameters. The coordinate-based method uses spatial limits (e.g., along x, y, or z) to assign materials, mirroring strategies used in multi-material additive manufacturing processes.

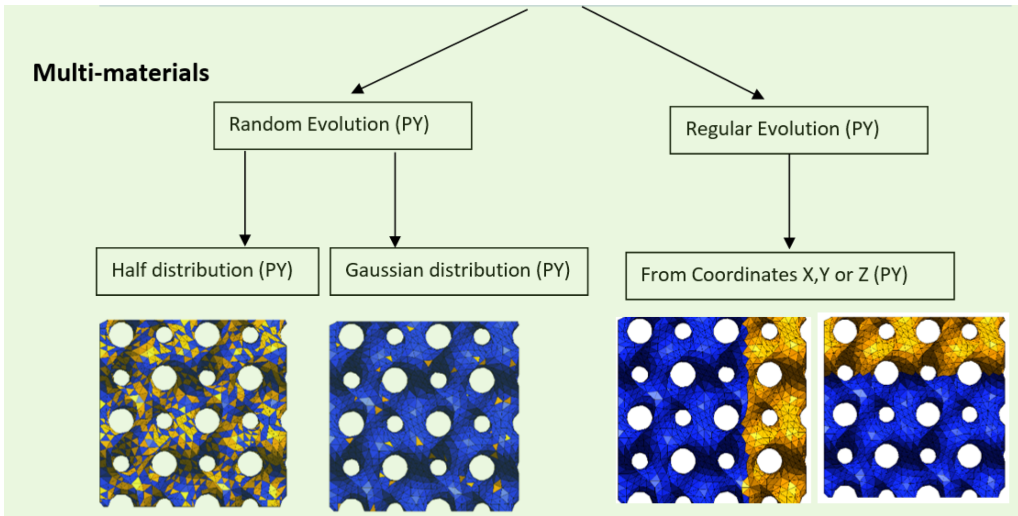

Fig. 6: Methodology for Multi-Material Lattice Design.

This flexibility enables users to create diverse material distributions for a single lattice design, significantly expanding the dataset. At the end of each generation session, the workflow outputs mesh files in a format compatible with Abaqus, supporting subsequent stages of analysis and optimization.

4. Conclusion

During this study, various lattice generation methods were reviewed, highlighting limitations in existing tools, particularly the inability to generate both truss-based and surface-based structures with multi-material capabilities. To address these gaps, a novel workflow was developed, integrating a graphical interface with Python-Grasshopper communication to enable efficient lattice generation and management of multi-material configurations.

The proposed approach facilitates the creation of diverse datasets suitable for AI-driven optimization. Key challenges, such as the generation of physically invalid lattice geometries (e.g., disconnected components), were systematically addressed through parameter validation and iterative refinement. Moreover, the export of lattices in STL format ensures compatibility with manufacturing processes, enabling the production of test specimens, including those intended for acoustic applications.

Currently, the methodology supports lattices with two distinct materials. Future developments could focus on expanding this capability to include three or more materials, allowing for more complex transitions and gradients, particularly in cases where physical properties at material interfaces are not yet well-defined. These advancements would further enhance the versatility of the workflow.

To build a comprehensive database for simulation and optimization, extensive lattice generations are planned, ensuring sufficient designs to support AI and numerical studies. Upcoming phases will focus on numerical simulations to optimize lattice designs and material distributions. Material data identification will play a crucial role in these simulations, while meshing remains a critical aspect for both finite element (FE) analysis and manufacturing. Investigations into the influence of mesh density and quality on FE responses will be essential to ensure accurate and efficient simulations.

Additionally, exploring homogenization techniques for lattice structures offers a promising avenue to reduce computational costs. By eliminating the need for detailed meshing in some cases, these techniques could streamline both simulation and optimization workflows, paving the way for faster and more efficient analyses.

Acknowledgements

This research was conducted as part of project MODEST in collaboration with ETH Zurich. The authors gratefully acknowledge the support and contributions provided by all partners involved in the project.

References

[1] J. Schmelzle, “(Re)Designing for Part Consolidation: Understanding the Challenges of Metal Additive Manufacturing,” *J. Mech. Des*., vol. 137, no. 11, pp. 111404, 2015.
[2] S. Yang, “A new part consolidation method to embrace the design freedom of additive manufacturing,” *J. Manuf. Process.*, vol. 20, pp. 444–449, 2015.
[3] M. Arredondo-Soto, “A review on tailoring stiffness in compliant systems, via removing material: Cellular materials and topology optimization,” *Appl. Sci*., vol. 11, no. 8, pp. 3538, 2021.
[4] J. Plocher, “Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures,” *Mater. Des.*, vol. 183, pp. 108164, 2019.
[5] M. Kadic, “3D metamaterials,” *Nat. Rev. Phys*., vol. 1, no. 3, pp. 198–210, 2019.
[6] K. Yu, “Magnetoactive acoustic metamaterials,” *Adv. Mater*., vol. 30, no. 21, pp.1706348, 2018.
[7] B. Jenett, “Discretely assembled mechanical metamaterials,” *Sci. Adv*., vol. 6, no. 47, pp. eabc9943.
[8] K. Hazlehurst, “Evaluation of the stiffness characteristics of square pore CoCrMo cellular structures manufactured using laser melting technology for potential orthopaedic applications,”. *Mater. Des.*, vol. 51, pp. 949-955.
[9] A. Chaudhari, “Experimental investigation of heat transfer and fluid flow in octet-truss lattice geometry,” *Int. J. Therm. Sci.*, vol. 143, pp. 64-75, 2019.
[10] P. Ekade, “Fluid flow and heat transfer characteristics of octet truss lattice geometry,” *Int. J. Therm. Sci.*, vol. 137, pp. 253-261, 2019.
[11] J. Broughton, “Comparison of single-phase convection in additive manufactured versus traditional metal foams,” *J. Heat Transf.*, vol. 142, no. 8, pp. 082201, 2020.
[12] M. I. Tenorio-Suárez, “MaSMaker: An open-source, portable software to create and integrate maze-like surfaces into arbitrary geometries,” *SoftwareX*, vol. 19, pp. 101203, 2022.
[13] O. Al-Ketan, “MSLattice: A free software for generating uniform and graded lattices based on triply periodic minimal surfaces,”. *Mater. Des. Process. Commun.*, vol. 3, no. 6, pp e205, 2021.
[14] F. Perez-Boerema, “A flexible and easy-to-use open-source tool for designing functionally graded 3D porous structures,” *Virtual Phys. Prototy.*, vol. 17, no. 3, pp. 682-699, 2022.
[15] I. Maskery, “FLatt Pack: A research-focussed lattice design program,” *Addit. Manuf.*, vol. 49, pp. 102510, 2022.
[16] A. Karakoç, “RegionTPMS—Region based triply periodic minimal surfaces (TPMS) for 3-D printed multiphase bone scaffolds with exact porosity values,” *SoftwareX*, vol. 16, pp. 100835, 2021.