Automated Numerical Workflow for Tensile Test Simulation of 3D-Printed Lattice-Reinforced Specimens

White paper

IT4Innovations

Ostrava, 2025

VSB-TUO, IT4Innovations National Supercomputing Center, 17. listopadu 2172/15**,** 708 00 Ostrava-Poruba**,** Czech Republic

Content

VSB-TUO, IT4Innovations National Supercomputing Center, 17. listopadu 2172/15**,** 708 00 Ostrava-Poruba**,** Czech Republic

1. Introduction

Additive manufacturing provides unique possibilities for tailoring the internal structure of components through a wide variety of lattice topologies, infill densities, and geometric configurations. These internal architectures make it possible to influence mechanical behaviour in ways that are not achievable with conventional monolithic materials.

To experimentally determine the tensile strength of 3D-printed specimens with lattice infills, one could conduct a systematic investigation into how different lattice types and infill densities affect the mechanical behaviour of PLA parts. Using an FDM printer and the wide selection of lattice patterns, it is possible to fabricate specimens with numerous internal architectures, each printed at several infill levels ranging from hollow (0%) to fully dense (100%). Such a design space would yield hundreds of samples suitable for tensile testing. The resulting measurements would provide a comprehensive dataset showing how internal geometry affects stiffness, load transfer, deformation localisation, and ultimate tensile strength.

Although the experimental campaign offers valuable insights, relying solely on physical testing limits the ability to analyse stress distributions, internal deformation modes, and the localised behaviour of struts within the lattice. Numerical simulation represents a powerful complementary tool that enables detailed, spatially resolved examination of stresses and strains across the entire specimen volume. With an adequate simulation framework, one can explore the influence of geometry with far greater granularity and isolate effects that are difficult to measure experimentally. However, the numerical modelling of such 3D-printed lattice structures poses several significant challenges.

The geometries exported directly from tools for 3D printing are not intended for engineering analysis; instead, they are optimised for toolpath generation. As a consequence, the exported mesh typically contains numerous defects such as intersecting surfaces, gaps between layers, non-manifold edges, open shells, and other inconsistencies that prevent direct meshing with volumetric finite elements. For finite element analysis, the geometry must be watertight and topologically clean, qualities that are rarely present in the raw models. This situation becomes particularly problematic for the internal lattices, where thin walls, sharp transitions, and small gaps are common, and where incorrect surface reconstruction can fundamentally alter mechanical behaviour.

The complexity of the problem grows further due to the scale of the experimental measurements. Because each lattice type is tested at many infill densities, hundreds of geometrically distinct

specimens must be prepared for simulation. Performing manual geometry repair for each variant is impractical. An automated or semi-automated approach is therefore essential, not only for efficiency but also to ensure consistency across the entire dataset. A robust workflow must be capable of importing raw generated geometries, correcting manifold issues, reconstructing continuous surfaces, and preserving both the outer shell and the intricate internal cavities that define the lattice behaviour. Once a valid geometry is obtained, it must then be converted into a high-quality tetrahedral mesh and processed using an appropriate numerical solver that can handle potentially enormous models with tens of millions of degrees of freedom.

The objective of this white paper is to design and demonstrate a workflow that utilises entirely open-source technologies where possible, supplemented by high-performance computing resources for the most demanding cases. The proposed workflow establishes a sequence that begins with geometry import and pre-processing, continues through voxel-based reconstruction using tools such as Blender [1] and the OpenVDB [2] library, and culminates in automated meshing via Gmsh [3] and a high-fidelity tensile simulation executed with the massively parallel ESPRESO [4] solver. The motivation is to show that, despite the significant geometric complexity and the challenges posed by slicer-based models, it is feasible to construct a scalable, repeatable, and computationally efficient pipeline capable of handling hundreds of specimen variants with minimal manual intervention.

The scope of this introduction, therefore, spans both the experimental context and the numerical challenges that motivate the development of the workflow. It outlines the necessity for watertight, accurate geometries; the difficulty of processing models with thin struts and small internal voids; the sensitivity of voxel-based remeshing to resolution; the computational demands imposed by high-density lattice structures; and the intrinsic need for automation. By situating the problem within this broader framework, it becomes clear why conventional CAD repair tools are insufficient and why carefully designed voxelisation and meshing procedures, coupled with HPC-enabled solvers, are essential to achieving credible and scalable numerical verification of the experimental tensile tests.

2. Geometry Processing Requirements

The numerical simulation of 3D-printed specimens with internal lattice structures depends critically on the quality and fidelity of the underlying digital geometry. Unlike traditional CAD models, which are typically constructed as idealised solids with mathematically well-defined surfaces, the geometries exported by slicers such as PrusaSlicer [5], see Figure 1, are derived from toolpath-generation logic rather than from watertight solid modelling principles. As a result, the raw geometry delivered for simulation rarely satisfies even the basic requirements of a manifold surface representation suitable for finite element analysis. The input specimens provided in this study clearly illustrate these issues:

although each model appears to represent a single printed part at first glance, closer inspection reveals that the exported mesh is composed of multiple independent surface fragments that frequently intersect one another, contain gaps, or fail to form a closed volume.

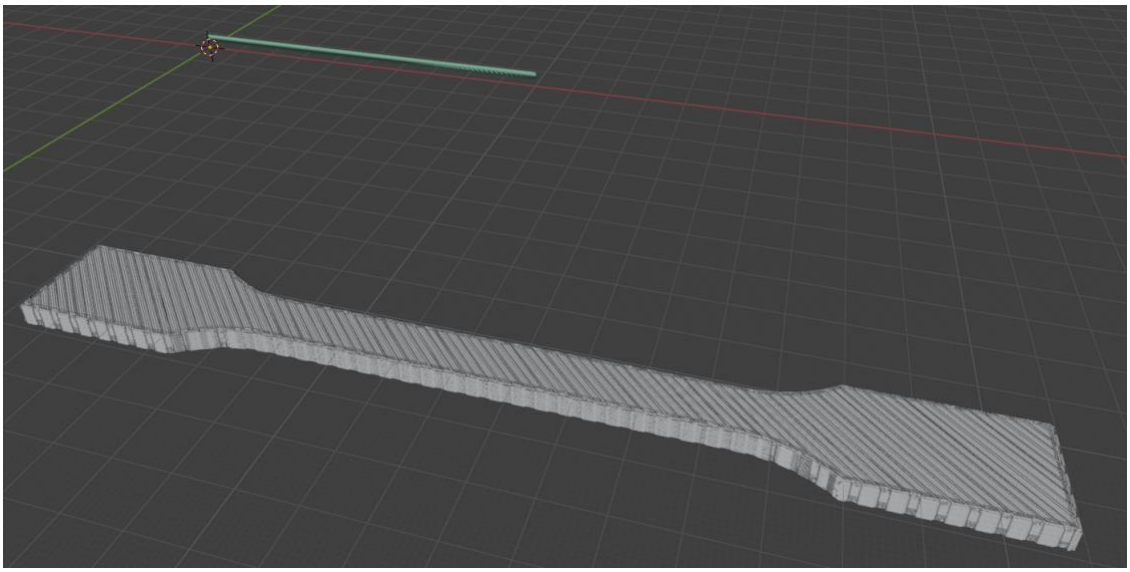

Figure 1 – Obtained geometrical model

This situation arises because the slicer reconstructs the part as a set of layered extrusions, where each layer is a collection of toolpath-defined surfaces representing the deposition of polymer material. The outer contour, the top and bottom surfaces, and the internal lattice are all generated as separate shell-like structures. Preferably, these components are delivered as four independent entities: the outer perimeter, the top skin, the bottom skin, and the internal lattice, see Figure 2. The interface between these components is not always well-defined: some surfaces penetrate others, some contain micro-gaps, and some only partially overlap due to numerical discretisation errors in the slicing process.

For simulation, however, a robust volume must exist. The generation of a finite element mesh from a surface representation requires that the surface form a coherent, watertight boundary that encloses a single contiguous region of space. Any topological irregularity, even a single missing triangle, can prevent meshing tools from identifying the interior region and constructing a consistent tetrahedral discretisation. Moreover, the quality of the geometry directly influences the quality of the mesh: intersections between surface patches create ambiguous boundaries; sharply angled or overlapping triangles degrade mesh quality; and extremely thin strut-like features impose strict requirements on element size, often leading to prohibitively large meshes if handled incorrectly. Thus, the initial preprocessing of geometry is not merely a convenience—it is a fundamental prerequisite for every subsequent stage of the workflow.

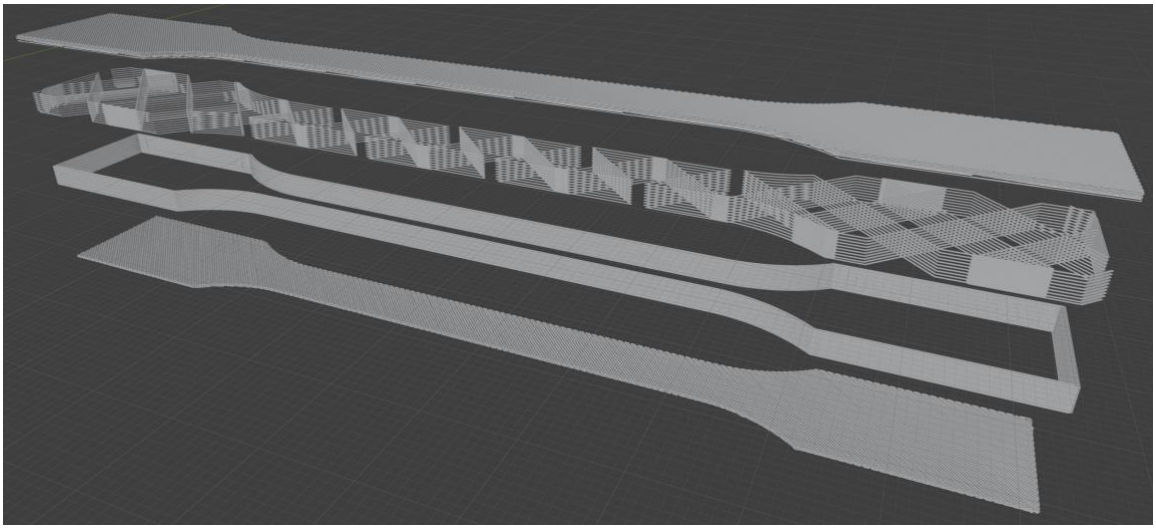

Figure 2 – Separated model components – from the top:
top skin, internal lattice, outer perimeter, bottom skin

The nature of the lattice structures compounds the problem. Lattices are inherently complex, consisting of numerous narrow channels, thin walls, and repeated patterns that must be preserved with high accuracy to maintain mechanical fidelity. Even a small defect in the reconstructed geometry can alter the connectivity or thickness of lattice elements, which in turn affects the predicted stress distribution, stiffness, and deformation behaviour. For lower-density lattices, these issues may be less severe, as the struts are well separated and the geometry is relatively open. However, as the infill density increases, particularly in the Cubic and Adaptive Cubic patterns, see Figure 3, entire regions of the lattice become enclosed by complex voids that are highly sensitive to reconstruction techniques.

Another central issue concerns the sheer scale of the models once they are prepared for meshing. Even when the geometry is successfully reconstructed, preserving thin features necessitates finely resolved surface meshes. High-density lattice samples may require voxel sizes on the order of 0.05 mm or less to capture all relevant details, which leads to surface meshes containing tens of millions of triangles. These extremely large meshes, while geometrically accurate, introduce substantial computational burdens at the meshing stage. Tools such as Gmsh can handle such data only with careful memory management and frequently require high-performance workstations or access to supercomputing resources. If the geometry contains any topological ambiguity, the meshing step may fail entirely, not due to the complexity of the shape, but because the watertightness requirement is violated somewhere within the massive surface dataset.

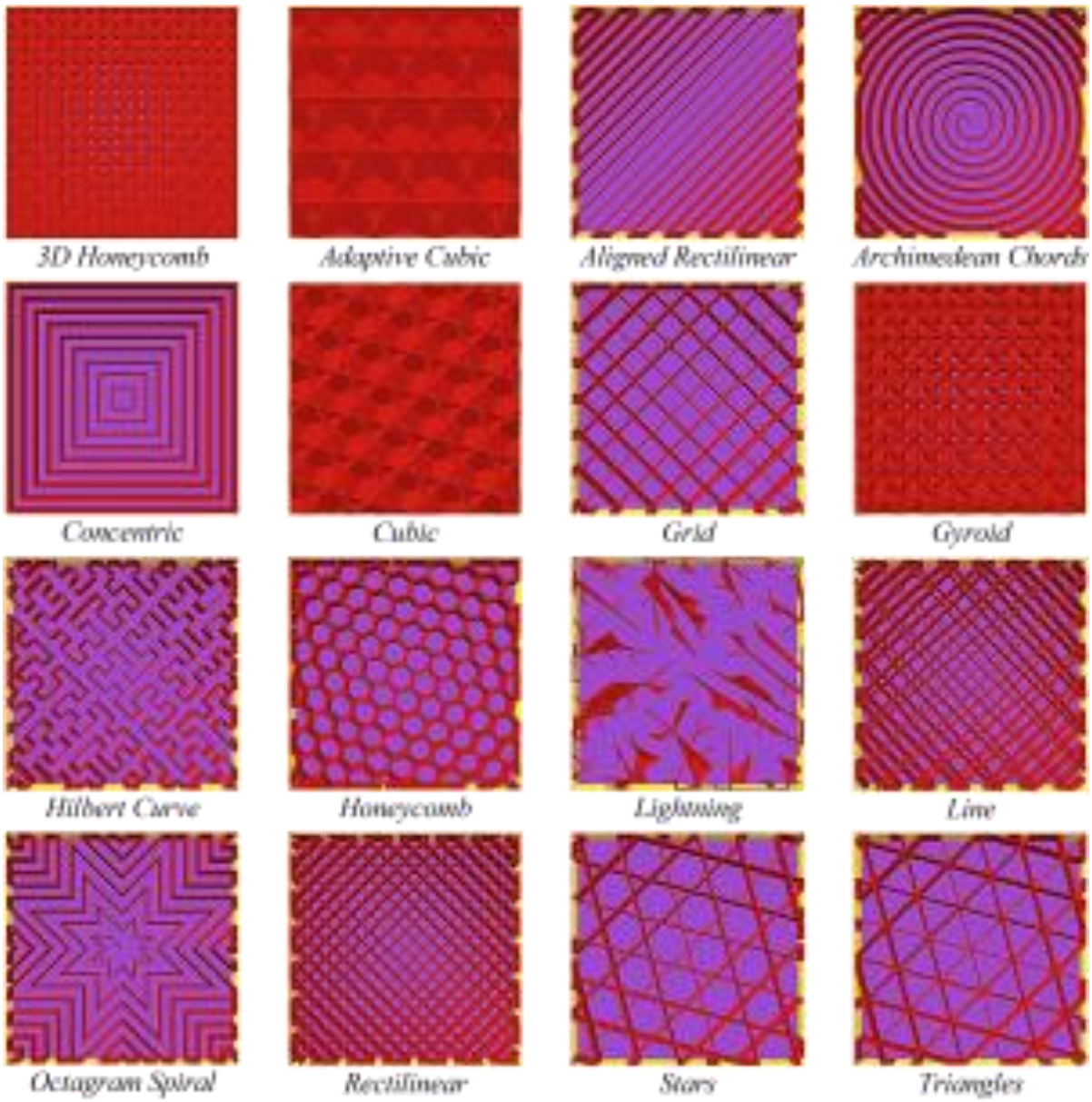

Figure 3 – 16 lattice patterns from the PrusaSlicer software

Therefore, the essential requirements for geometry processing in this workflow are twofold. First, the geometry must be transformed from its problematic slicer-exported form into a closed, manifold, watertight surface that accurately represents both the outer shell and the intricate internal lattice. This transformation must be performed in a manner that preserves key geometric details while eliminating defects such as gaps, intersections, and floating surfaces. Second, this process must be automatable, as we are dealing with hundreds of distinct specimens, each with a unique internal structure. The traditional manual repair techniques used in CAD environments, while sometimes effective for small numbers of models, are wholly inadequate for processing large datasets of this type. Automation is

not only a matter of efficiency but also of reproducibility: the pre-processing steps must yield consistent results regardless of the specific lattice pattern or infill density.

The remainder of the workflow rests entirely on the fulfilment of these requirements. Once a watertight geometry is available, mesh generation, boundary-condition assignment, and numerical simulation become straightforward technical operations. Without such a geometry, however, no amount of solver sophistication can compensate for topological errors introduced at the geometry stage. The geometry processing requirements thus define the foundation of the entire simulation pipeline and motivate the exploration of voxelization-based reconstruction methods and high-fidelity libraries, such as OpenVDB, which can process complex internal structures in a reliable and scalable manner. As the following sections show, meeting these requirements is both the most challenging and the most critical component of the proposed automated simulation workflow.

3. Evaluation of Geometry Reconstruction Methods

The geometry preprocessing stage in the simulation pipeline revolves around reconstructing a watertight and topologically valid representation of the specimen from the raw mesh fragments exported by PrusaSlicer. Because the slicer-generated geometry is fundamentally unsuited for direct finite element meshing, several reconstruction strategies were evaluated to determine their suitability for automation, their ability to preserve fine lattice features, and their robustness across a wide range of infill densities and lattice topologies.

3.1 Limited Geometry Repair in Ansys SpaceClaim

When preparing lattice-based geometries for simulation, Ansys SpaceClaim [6] can be a helpful tool, but it must be used with a clear understanding of its capabilities and limitations. SpaceClaim works well when the input geometry is already close to a clean, watertight solid. In such cases, its automated repair functions and scripting capabilities can efficiently resolve minor defects and support large-scale batch processing. However, slicer-generated geometries present a fundamentally different challenge. Instead of representing true physical boundaries, slicers export fragmented and intersecting surface patches that approximate extrusion toolpaths, see Figure 4. These disconnected surfaces do not form a coherent shell, and as a result, SpaceClaim's stitching and solid-repair operations typically fail because the software has no topological basis on which to reconstruct a closed volume.

The Shrinkwrap tool may appear to offer a workaround, since it can generate a continuous envelope by effectively draping a surface over the input mesh. While this approach can succeed for simpler lattice types, it inevitably smooths or seals narrow passages and internal features. Fine lattice members may be merged or removed entirely, and dense infill structures often lose their internal cavities. Because these internal voids and connections are essential for accurately capturing stiffness, load

transfer, and deformation mechanisms, any modification of this topology leads to a model that no longer represents the printed specimen's true mechanical behaviour. For simulations where the internal architecture drives the structural response—such as tensile testing of lattice-infill specimens—these alterations make the geometry unsuitable for credible analysis.

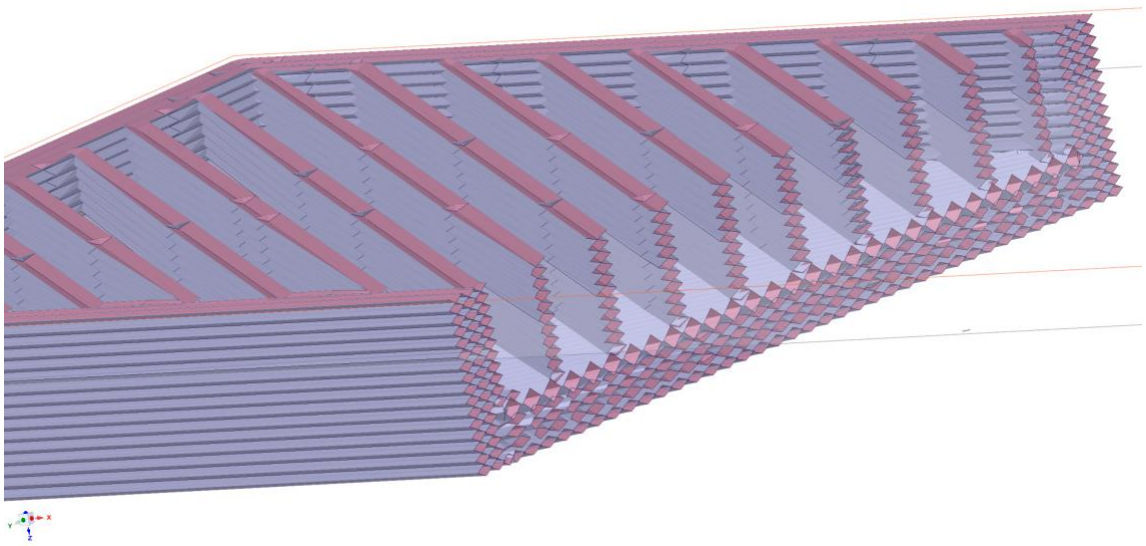

Figure 4 – Longitudinal and transverse section of the obtained model with an Aligned Rectilinear lattice

For these reasons, SpaceClaim should not be relied upon as an automated repair tool for complex slicer-generated lattices. Its strength lies instead in diagnosing geometric issues, visualising defects, and experimenting with potential preprocessing strategies. Although it can provide valuable insight into the challenges of surface reconstruction, more specialised tools are required when the accurate preservation of internal topology is essential.

3.2 Blender Voxel Remeshing

Blender is a highly flexible open-source geometry environment with a powerful Python API that allows complete automation of geometry transformations. Blender offers several geometry modifiers capable of reconstructing surfaces from irregular meshes, but the most promising for this application is the Voxel Remesh modifier. Unlike direct surface-based repair tools, voxel remeshing converts the geometry into a volumetric representation before extracting a new isosurface. This approach is well-suited to handling fragmented or non-manifold meshes because voxelization inherently ignores the original topology and instead reconstructs the solid based on the occupancy of voxels.

The underlying principle of voxel remeshing is conceptually straightforward: the geometry is embedded in a 3D voxel grid, where each voxel is assigned a binary or signed-distance value indicating whether it lies inside or outside the model. Once this volumetric representation is generated, Blender

reconstructs a new surface mesh by extracting an isosurface through the voxel field. The resulting mesh is guaranteed to be watertight and free of topological inconsistencies. Because the method is geometry-agnostic, it handles intersecting surfaces gracefully and eliminates gaps automatically.

However, this robustness comes at the cost of sensitivity to voxel resolution. The voxel size defines the minimum feature size that can be preserved during reconstruction. Coarse voxel resolutions (e.g., 0.2 mm) are adequate for capturing the general shape of low-density lattices, where struts are separated by sufficiently large gaps. In such cases, the remeshed surface provides a reasonable approximation of the specimen's internal geometry with manageable mesh sizes. For denser lattice structures, however, the gaps between adjacent lattice members become comparable to or smaller than the voxel size. When the voxel grid cannot resolve these gaps, adjacent struts merge, blurring the lattice structure and altering its mechanical topology.

Moreover, Blender's voxel remeshing algorithm presents challenges for lattice structures due to two distinct error mechanisms: resolution dependency and topological occlusion. While coarse voxel grids fail to resolve intricate voids due to aliasing, a more critical issue arises from the algorithm's Signed Distance Field (SDF) calculation. If a lattice structure contains sealed internal cavities, as in the case of Cubic and Adaptive Cubic lattices at high infill densities, the SDF generation frequently classifies the enclosed space as 'inside' the mesh. This results in the complete filling of internal voids regardless of voxel size, producing a solid block rather than a porous structure, see Figure 5. As this compromises both geometric accuracy and the mechanical validity of subsequent simulations, Blender's voxel remeshing is often unsuitable for high-density lattice generation.

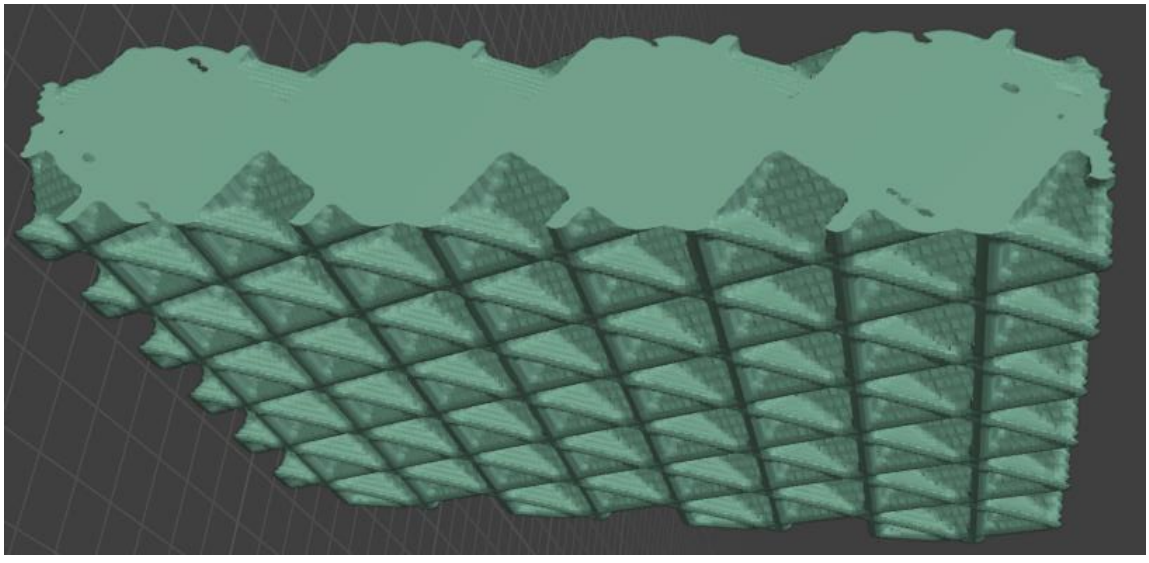

Figure 5 – Transverse section of 70% Adaptive Cubic lattice after Remesh with 0,1 mm voxel size

Despite these challenges, Blender remains an essential component of the workflow due to its complete scriptability, ease of integration with Python, and strong ability to handle low- and medium-density lattice structures. For many of the lattice configurations tested in this project, Blender's Voxel Remesh produced robust, watertight geometries while maintaining the essential features of the original lattice. Its flexibility makes it an ideal first-stage tool in the computational pipeline.

3.3 High-Fidelity Voxelization with OpenVDB

OpenVDB represents a class of volumetric data structures specifically engineered to handle large, sparse voxel grids efficiently. Originally developed for high-end visual effects and animation, OpenVDB offers a hierarchical, memory-efficient representation that stores voxel data only where necessary, allowing for the processing of extremely high-resolution volumes without the prohibitive memory footprint associated with uniform grids.

OpenVDB's approach differs significantly from Blender's voxel remeshing. Rather than producing a simplified or smoothed representation of the input geometry, OpenVDB maintains high-fidelity detail by capturing surface information through signed-distance fields. As shown in Figure 6, this allows the tool to accurately retain both thin lattice struts and the internal cavities that arise, especially in the case of Cubic and Adaptive Cubic lattices of higher infill densities. Because of its sparse representation, OpenVDB can manage extremely fine voxel resolutions—far beyond what Blender can handle—without generating unmanageable file sizes or overwhelming system memory.

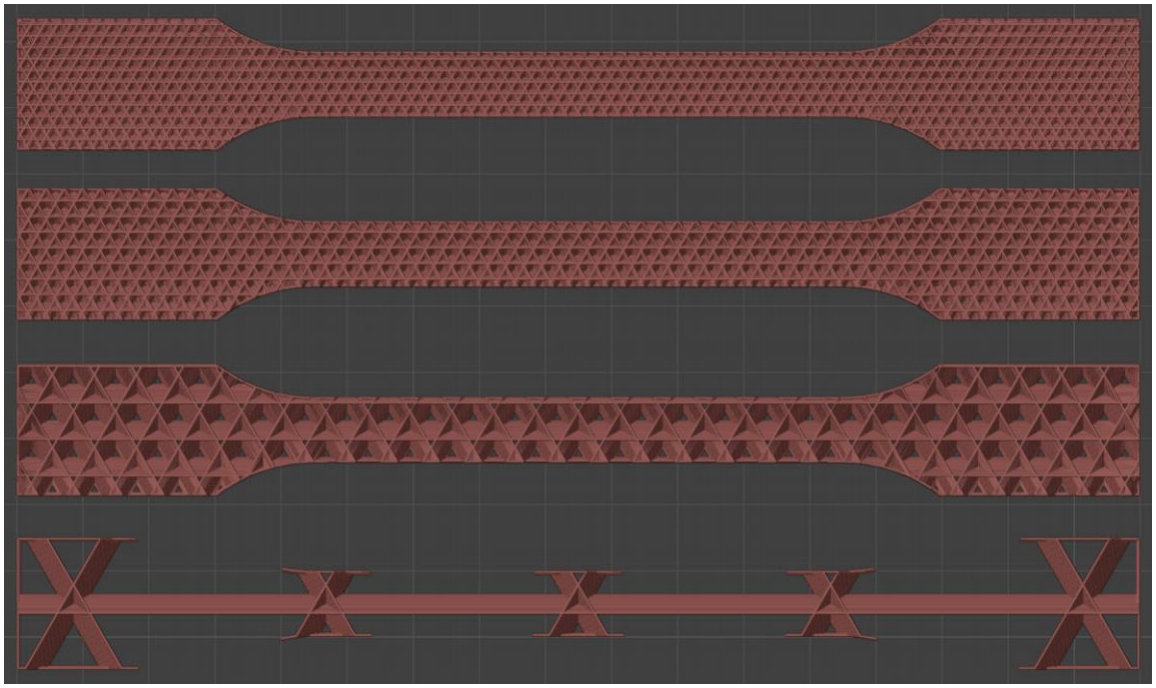

Figure 6 – Lattice Adaptive Cubic with infill densities:
from the top 95 %, 70 %, 30 % and 5 %

When reconstructing a lattice model, OpenVDB effectively voxelises the geometry at user-defined resolution, applies optional morphological filters or smoothing operations if needed, and finally extracts a manifold surface using a high-precision isosurface extraction algorithm. The resulting STL files preserve both external geometry and complex internal void networks with far greater consistency than Blender, particularly in the presence of deeply nested cavities and tight clearances. This fidelity makes OpenVDB especially valuable when simulating high-density lattice structures, where traditional remeshing tools fail to maintain topological correctness.

The trade-off, however, lies in implementation complexity. Unlike Blender, which offers a graphical user interface and a Python API suitable for immediate experimentation, OpenVDB requires direct manipulation of volumetric data structures through Python bindings or C++ interfaces. Setting up the workflow involves integrating libraries, configuring voxel grids, performing distance-field operations, and ensuring compatibility with downstream mesh generators such as Gmsh. Nevertheless, once configured, the approach is entirely scriptable and offers unmatched robustness for geometries that would otherwise be impossible to prepare.

In this sense, OpenVDB complements Blender: where Blender provides ease of use and automation for low and medium densities, OpenVDB fills the high-fidelity requirements of the most complex cases if they need to be simulated.

4. Automated Mesh Generation Workflow

Once a watertight, topologically consistent surface representation of the specimen has been reconstructed, whether through Blender's voxel remeshing or the higher-fidelity OpenVDB workflow, the next fundamental step in the numerical pipeline is the creation of a volumetric finite element mesh. The quality and robustness of this mesh have a decisive influence on the fidelity and stability of the subsequent computational analysis. For lattice-reinforced specimens, the meshing task is exceptionally demanding: the internal architecture consists of thin struts, narrow channels, and highly variable cross-sections that must be discretized sufficiently finely to capture the deformation and stress distribution accurately. At the same time, the global size of the model must remain manageable, and the entire process must be automated to accommodate the hundreds of variants produced experimentally.

The challenge of meshing these geometries begins with the nature of the surface mesh generated during the reconstruction phase. As documented in prior sections, voxel-based reconstruction, especially when performed at fine resolutions, is capable of producing extremely detailed surface meshes containing tens of millions of triangles. Such high-resolution surfaces are essential for

representing dense lattice structures but impose significant computational burdens on meshing algorithms. In particular, the meshing tool must interpret the triangulated manifold surface as a coherent closed boundary that encloses a well-defined interior. Modern meshing libraries, including Gmsh, are highly capable, but they still require surface meshes that meet strict topological criteria: every edge must be shared by exactly two faces; no triangles may intersect or overlap improperly; and the surface must form one or more complex but closed loops defining a consistent interior region. Any deviation from these conditions introduces ambiguity into the notion of “inside” and “outside,” preventing volume generation.

In conventional CAD systems, such errors might be resolved manually through topology healing tools; however, manual repair is incompatible with a workflow that must scale to hundreds of inputs. It became clear that geometry reconstruction methods must produce not only a watertight model but also a geometrically clean one, free from spurious elements. The shift to voxel-based reconstruction largely resolved this issue, as voxelisation inherently eliminates these inconsistencies by replacing the original fragmented geometry with an entirely new isosurface.

With a valid watertight surface available, meshing proceeds by importing the STL file into Gmsh, which interprets the surface as a “discrete mesh”, a representation of the boundary without any underlying CAD entities. In this mode, Gmsh first attempts to identify the closed surface loops that define enclosed volumes. For simple geometries, this detection is straightforward. For lattice structures, however, the mesher must navigate thousands or millions of tiny, interconnected triangles forming highly convoluted shells. Despite this complexity, voxel-based surfaces are generally well suited to automated loop detection because the reconstructed surface contains no open edges or ambiguous boundaries. Nevertheless, the extremely high resolution of these surfaces remains a practical challenge. Memory consumption increases substantially with mesh size, and the initialisation of the meshing algorithm can take a considerable amount of time. In those cases, meshing could be feasible only on high-performance computing nodes with large memory capacity, as typical workstations lack sufficient RAM.

Another important consideration is the size of the tetrahedral elements used to fill the interior. Ideally, the element size should be fine enough to resolve the thickness of the lattice walls while remaining coarse in regions containing solid material, such as the outer grips of the specimen. Gmsh supports several strategies for controlling the characteristic length of elements, including global parameters and user-defined fields that vary spatially. In practice, however, applying nonuniform refinement to a model containing thousands of thin interlocking cells is nontrivial, and global refinement is often the only reliable approach. Consequently, many meshing runs could result in volumetric meshes with millions to hundreds of millions of tetrahedral elements. Even in the most efficient implementations, generating such large meshes can take hours and require substantial computational resources.

During meshing, two principal limitations could emerge. The first is the oversized input surface mesh: Blender-generated surfaces for 80% and higher-density lattice structures frequently contain upward

of 40 million nodes when using voxel sizes of 0.05 mm. Importing such massive STL files into Gmsh is possible but slow, and the subsequent volume meshing step often becomes the computational bottleneck of the entire workflow. Some mitigation strategies were explored, including mesh decimation in Blender and level-of-detail reductions in OpenVDB, but these introduce trade-offs between geometric fidelity and computational feasibility. For geometries where internal cavities define crucial mechanical behaviour, excessive decimation may compromise simulation accuracy, and thus careful balancing is required.

The second limitation relates to the necessity of completely watertight and clean surfaces. Even after voxel-based reconstruction, occasional isolated artefacts or degenerate triangles can persist, especially in regions where the voxel grid skirts ambiguous boundaries. These artefacts can disrupt volume definition. Gmsh, equipped with the OpenCASCADE kernel, includes routines for geometry repair, but such operations are not always effective on STL meshes because they lack explicit topological or parametric structure. Instead, the most reliable solution proved to be iterative refinement of the voxelisation stage itself, adapting voxel resolution, revising smoothing operations, and fine-tuning geometry reconstruction, to ensure that the surface produced is inherently valid and requires no downstream repair.

Despite these challenges, once a valid volumetric mesh is generated, the remaining steps in the workflow become relatively straightforward. Tetrahedral meshes produced by Gmsh can be exported directly in a format compatible with the ESPRESO solver. The uniformity and quality of the mesh are especially crucial for parallel performance, as poorly shaped elements or inconsistencies in domain decomposition can significantly impair scalability. The consistency of the reconstructed geometry from sample to sample also enabled automated assignment of boundary conditions, as the positions of the gripping regions remained identical across all variants.

Ultimately, the automated meshing workflow described here demonstrates that generating FEM meshes for dense, complex lattice structures is feasible but requires both strategic preprocessing and substantial computational resources. The transition from fragmented slicer-based geometries to high-fidelity voxelised surfaces is essential to achieving robustness, and the use of Gmsh allows complete automation once a watertight STL is available. Nevertheless, the meshing stage remains one of the most computationally demanding components of the entire pipeline, often necessitating HPC-level memory and processing power. These considerations underscore the importance of efficient geometry reconstruction techniques and highlight the close interdependence between preprocessing choices and the tractability of downstream simulation tasks.

5. Numerical Simulation of Tensile Testing

With the volumetric mesh successfully generated, the workflow advances to its final and most computationally demanding stage: the finite element simulation of a tensile test. The purpose of this stage is to numerically replicate the mechanical response typically observed in the experimental testing and to provide insights into the internal stress fields, deformation modes, and load-transfer mechanisms that cannot easily be observed through physical testing alone. The simulation environment must therefore be capable of handling extraordinarily large models while ensuring numerical stability, accuracy, and compatibility with automation.

At the core of the simulation lies the material model. For PLA specimens subjected to monotonic tensile loading, the mechanical response up to the onset of yielding can be approximated using linear elasticity. This approximation is justified because the primary objective of this study is not to capture failure or large-strain plasticity, but rather to quantify the stiffness characteristics and early-stage stress distribution in lattice-reinforced specimens. Linear elasticity requires only two material parameters, the Young's modulus and the Poisson ratio, making it straightforward to apply consistently across the entire dataset. Although this simplification inherently limits the simulation's ability to predict ultimate tensile strength or post-yield behaviour, it allows for a highly robust and computationally efficient analysis suitable for processing hundreds of samples.

Boundary conditions form the second essential component of the simulation setup. To ensure consistency between numerical and experimental conditions, the boundary configuration mirrors that used during the physical tensile tests. In all specimens, the left end of the sample is fully constrained, corresponding to a fixed grip in the testing machine. All translational degrees of freedom are zeroed in this region, preventing any movement or rotation. On the right end, a prescribed displacement is applied in the axial direction, simulating the controlled extension imposed by the tensile-testing apparatus. This displacement-driven formulation is advantageous because it avoids numerical difficulties associated with force-driven loading in extremely compliant structures such as low-density lattices. The uniform geometry across all specimens, each with identical grip geometry and position, greatly simplifies the automation of boundary-condition assignment. The solver can identify the relevant nodes based on their spatial coordinates, enabling a fully parameterised and reproducible setup free from manual intervention.

Once the boundary conditions and material parameters are defined, the simulation must be executed using a solver capable of processing meshes containing hundreds of millions of elements. For this purpose, the workflow employs the ESPRESO solver, a massively parallel finite element framework developed specifically for high-performance computing environments. ESPRESO's architecture is designed around scalable domain decomposition techniques that distribute the computational workload across many processors, enabling the efficient solution of extremely large linear systems. In this context, the size of the volumetric meshes, particularly those associated with high-density lattice specimens, makes parallel execution indispensable. Serial solvers or standard workstation-level FEM tools cannot reliably handle such models due to memory constraints, excessive runtime, or both.

During test simulations conducted on representative specimens, ESPRESO demonstrated strong robustness and scalability. The solver efficiently partitioned the domain into subregions assigned to different compute nodes, allowing the tensile problem to be solved within practical timeframes even for models with hundreds of millions of degrees of freedom. In particular, the linear elastic formulation is well suited to parallelisation, as it avoids the iterative nonlinear updates required by plasticity or large-deformation kinematics. As long as the mesh quality remains high, which is generally ensured by the voxel-based geometry reconstruction and Gmsh meshing pipeline, the numerical conditioning of the stiffness matrix remains favourable, contributing to stable and reliable convergence.

The simulation results provide valuable insights into the structural behaviour of lattice-reinforced specimens. The displacement fields reveal the global deformation pattern, illustrating how the lattice structure accommodates elongation and distributes strain. Stress visualisations, especially equivalent von Mises stress fields, illuminate regions of stress concentration along struts, junctions, and boundary interfaces. For low-density lattices, stress tends to localise at thin strut intersections or areas where the load path is discontinuous, reflecting the inherent structural heterogeneity of sparse infill patterns. High-density lattices, by contrast, exhibit more uniform stress distributions, with fewer localised peaks but overall higher stiffness. The results reproduced and extended observations made experimentally, providing a detailed internal perspective inaccessible to physical instruments. Figure 7 shows characteristic stress patterns in a Rectilinear 5% infill specimen, clearly illustrating how struts near the load path carry the majority of the stress while peripheral regions deform more freely.

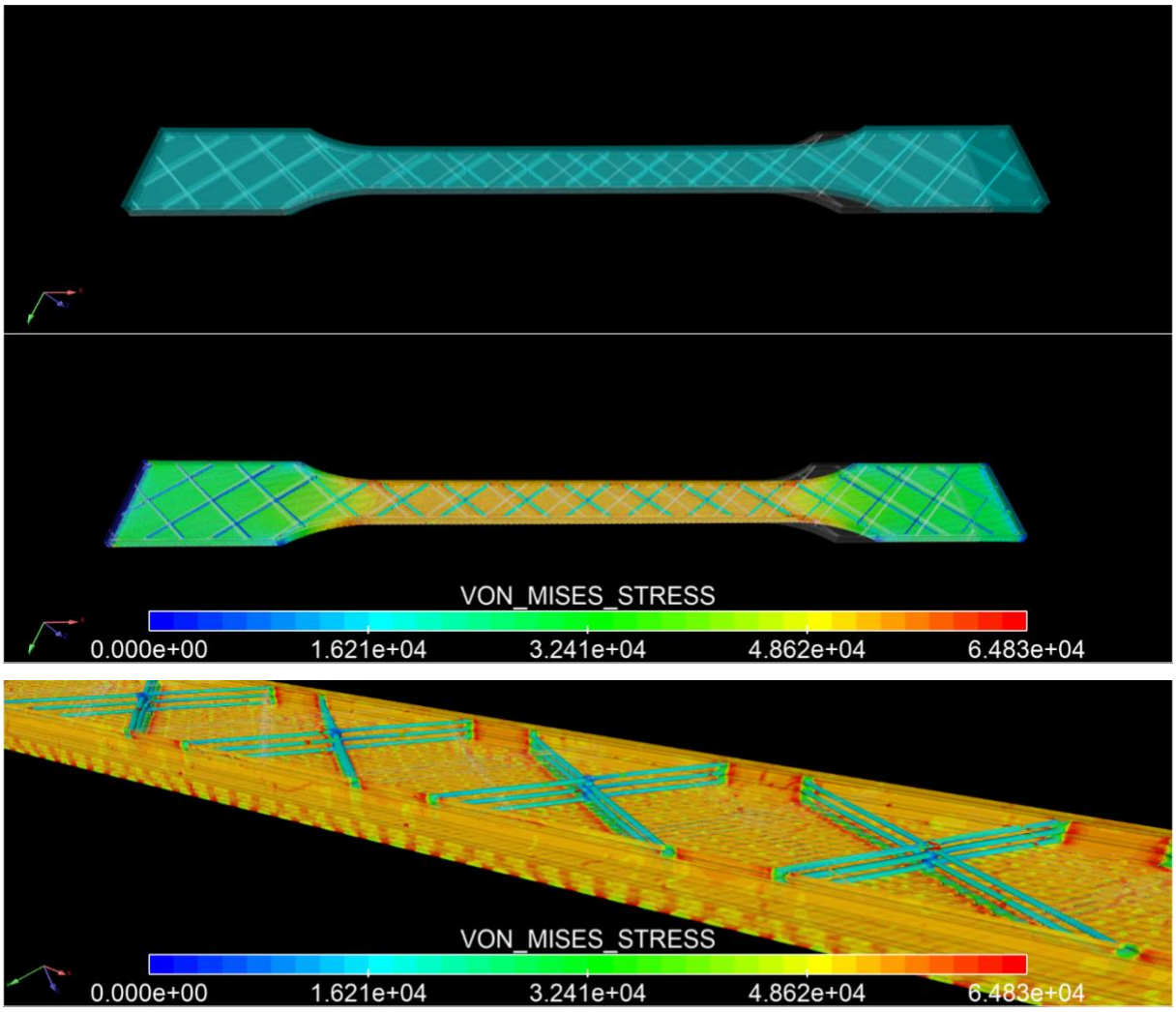

Figure 7 – Equivalent von Mises stress in MPa displayed on the deformed specimen with 5% Rectilinear lattice

An important dimension of the simulation process concerns computational efficiency and data management. The largest simulations require substantial amounts of memory, particularly when assembling stiffness matrices for finely meshed lattice structures. For dense geometries, even storing the volumetric mesh and associated solver data structures can exceed the memory available on conventional machines. Therefore, simulations were executed on a supercomputing infrastructure, allowing the use of distributed memory and parallel processing. This reliance on HPC resources underscores the unique challenge posed by lattice structures: their geometric complexity fundamentally drives computational cost, making efficient solvers and large-scale parallelism essential. ESPRESO's strong performance in this respect validates its integration into the workflow.

Finally, it is worth noting the limitations inherent to the linear elastic model employed. While linear elasticity captures the initial stiffness and elastic deformation behaviour, it cannot predict yield, plastic flow, or failure, which are crucial for evaluating ultimate tensile strength. These behaviours become particularly relevant for certain lattice types that exhibit buckling, local collapse, or progressive failure under tension. Extending the workflow to include nonlinear material models or fracture analysis would enable such predictions but would introduce significant challenges, including higher computational cost, potential convergence difficulties, and the need for accurate material calibration. Such extensions are feasible but lie beyond the current scope of this study.

In its present form, the simulation stage of the workflow demonstrates that large-scale numerical analysis of lattice-reinforced additive-manufactured specimens is both achievable and reliable, provided that the geometry and mesh generation pipelines deliver high-quality inputs. By coupling voxel-based geometry reconstruction with high-performance finite element computation, the workflow enables a deeper understanding of the mechanical behaviour of complex lattice structures and provides a solid numerical counterpart to experimental tensile testing.

6. Results and Discussion

The culmination of the preprocessing, meshing, and simulation stages is a set of numerical results that can be compared with experimental observations and evaluated for internal consistency, mechanical plausibility, and computational performance. Because the geometry reconstruction methods varied in their fidelity and robustness across the wide spectrum of lattice densities and topologies, one major focus of the analysis concerns the strengths and limitations of each approach when applied to practical simulation workflows. Another focus is the interpretation of the mechanical response predicted by the finite element models and how these results align with expectations derived from the printed specimens' physical behaviour.

The first major outcome concerns the quality of geometric reconstruction. Across hundreds of candidate geometries, Blender's voxel remeshing proved highly effective for low- and medium-density lattice structures, reliably producing watertight surfaces while preserving the major geometric features that govern mechanical behaviour. For infill levels up to roughly 30 %, the voxel size required to resolve geometrical features remained moderate (approximately 0.2 mm), see Figure 8 and Figure 9, and the resulting meshes were computationally tractable. As shown in the Figure 10, even at intermediate densities near 50 %, Blender produced valid geometries when the voxel resolution was reduced to 0.1 mm, although the resulting surface mesh consists of more than 5 million nodes, which increased the computational cost notably.

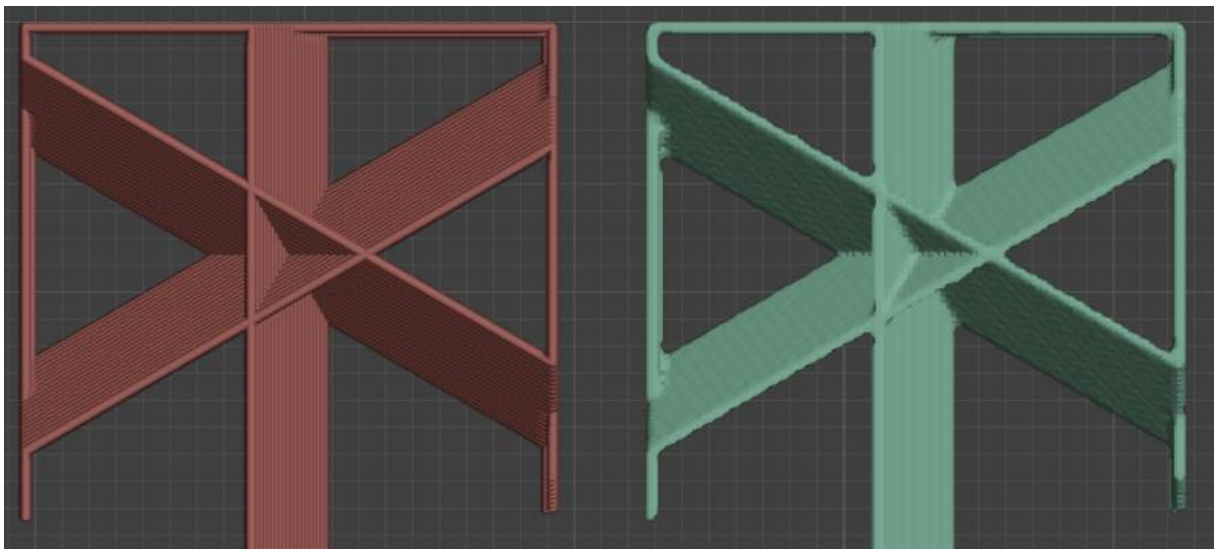

Figure 8 – 5% Adaptive Cubic lattice – from left: original, voxel size 0.2 mm

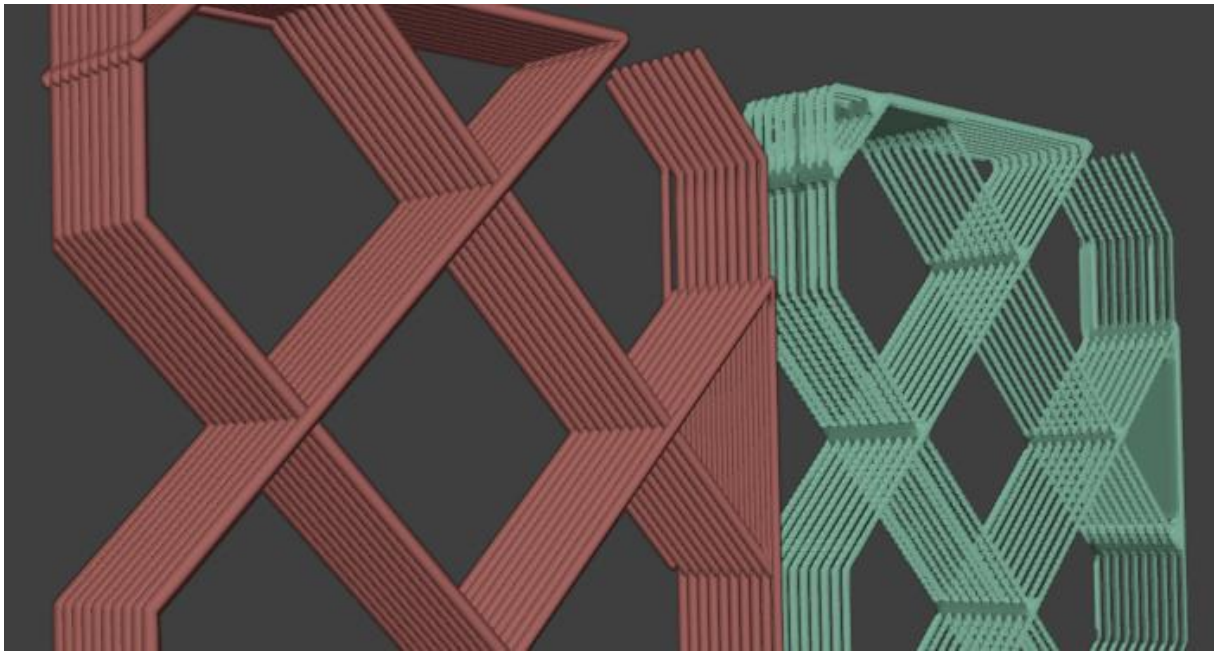

Figure 9 – 5% Rectilinear lattice – from left: original, voxel size 0.2 mm

In contrast, high-density structures, particularly those derived from Cubic and Adaptive Cubic lattice patterns, exhibited strong dependency on resolution and revealed the inherent limitations of Blender's remeshing algorithm. As shown in the Figure 11, at densities approaching 80 % or higher, maintaining correct internal cavities required voxel resolutions as small as 0.05 mm, leading to surface meshes with tens of millions of nodes. These fine resolutions were necessary to prevent the unwanted filling of

internal voids or merging of adjacent lattice members, both of which fundamentally alter the mechanical topology of the parts. However, the computational burden associated with such fine voxel sizes became a practical bottleneck, at times exceeding the memory limits of conventional workstation environments. These findings emphasise that while Blender offers an excellent, user-friendly platform for automated reconstruction of moderately complex lattice structures, it cannot be expected to handle the most intricate topologies at extreme densities without substantial performance trade-offs.

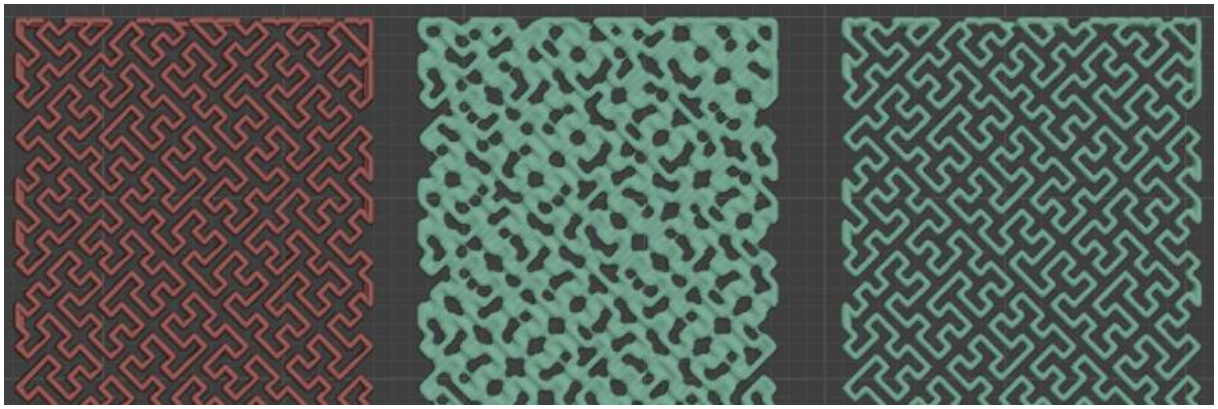

Figure 10 – 50% Hilbert Curve lattice – from left: original, voxel size 0.2 mm, voxel size 0.1 mm

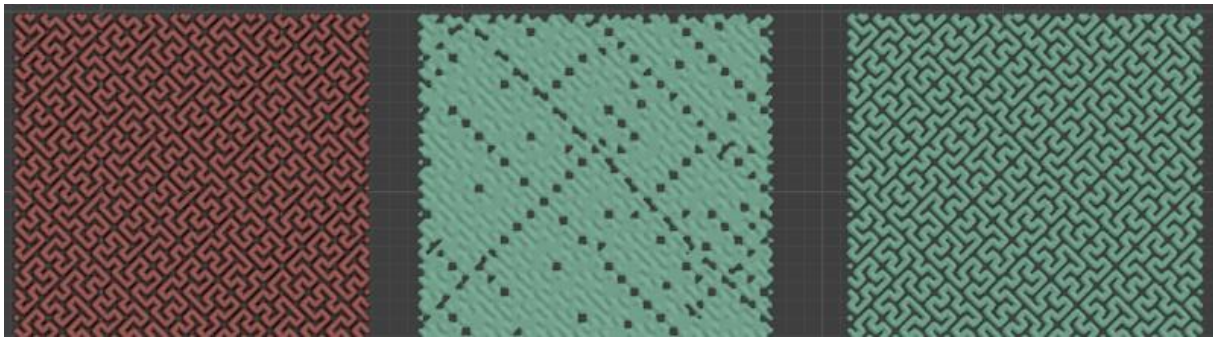

Figure 11 – 80% Hilbert Curve lattice – from left: original, voxel size 0.1 mm, voxel size 0.05 mm

OpenVDB filled this gap by providing a high-fidelity volumetric reconstruction method capable of preserving fine geometric details that Blender suppressed or collapsed. For the challenging high-density cases, OpenVDB consistently produced surface meshes that accurately retained internal cavities and thin strut geometries, even when those features approached the limits of the slicer's resolution. This fidelity is essential for producing meaningful mechanical simulations, as even small deviations in internal geometry can dramatically alter the predicted stiffness and localized stress distribution. OpenVDB's sparsity-aware data structures enabled the use of higher voxel resolutions than Blender could manage, making it the only viable reconstruction tool for certain categories of

specimens. The trade-off was the increased complexity of implementation and the need for more specialised software integration, but the robustness gained justified this investment, particularly in applications where geometric precision is paramount.

Beyond geometric fidelity, another dimension of the results concerns the generation, and quality of volumetric finite element meshes. The meshing process introduced several computational constraints, particularly for high-density geometries that required extremely fine surface resolutions. The time and memory required for meshing grew significantly with mesh size, occasionally leading to situations where only high-memory HPC nodes could complete the process. Yet, despite these challenges, the tetrahedral meshes generated by Gmsh were generally of high quality, without the distorted or degenerate elements that can lead to numerical instability. This stability reflects the benefits of voxel-based reconstruction, whose inherently smoother, more regular surfaces lend themselves well to meshing algorithms.

From the perspective of mechanical response, the linear elastic simulations produced results that align closely with physical intuition. Low-density samples exhibited highly localised deformation patterns, with strain concentrating at narrow lattice junctions or thin struts aligned with the load path. Stress distributions were correspondingly uneven, showing clear hotspots in regions where structural connectivity was weakest. Medium-density lattices showed more distributed load-sharing, with stress patterns reflecting the interplay between load-bearing pathways and structural redundancy. High-density samples tended to behave more like solid bodies, with more uniform stiffness and fewer localised stress concentrations. These numerical observations correlate well with experimental trends previously reported in the literature and with the physical behaviour observed during tensile testing.

While the present simulations did not attempt to estimate ultimate tensile strength or track material failure, the displacement and stress fields obtained nevertheless illuminate the structural mechanisms that shape macroscopic mechanical behaviour. The ability to visualize internal stress states provides insights that would be difficult to derive from experiments alone—particularly in low-density lattice structures, where internal deformation occurs in complex and spatially heterogeneous patterns.

The primary limitation of the current results is the adoption of a purely linear elastic material model. For many practical applications, including the study of failure modes, buckling of slender struts, or post-yield deformation, a nonlinear model would be required. However, given the size and complexity of the models in this study, a nonlinear approach would impose significantly greater computational costs and solver challenges, likely requiring additional HPC resources. Nonetheless, the linear elastic results obtained here provide a solid foundation for validating the geometry and mesh generation pipeline and for characterizing the early-stage mechanical behaviour of the specimens.

In summary, the results confirm both the feasibility and the limitations of large-scale automated simulation workflows for lattice-filled additive-manufactured specimens. Blender-based voxel

remeshing is effective for most geometries, but OpenVDB becomes indispensable at the high end of the density spectrum. Gmsh successfully converts reconstructed surfaces into high-quality tetrahedral volumes, albeit with substantial memory demands. ESPRESO performs well on these large meshes, providing reliable mechanical predictions that align with experimental trends. Together, these components constitute a robust and scalable framework for numerical analysis of complex lattice structures.

7. Automation Workflow Design

While the preceding sections describe the individual components of the geometry reconstruction, meshing, and simulation processes, the practical value of this methodology emerges most clearly when all these steps are combined into a cohesive, fully automated workflow. Such automation is essential for handling the scope of this project, which involves hundreds of specimen variants, each with unique internal geometries and requiring identical processing steps. Automation ensures consistency, reproducibility, and efficiency, minimizing the potential for human error and enabling the workflow to be executed with minimal supervision.

The workflow begins with the import of raw slicer-generated geometries, typically stored in OBJ format. Because these geometries are highly fragmented and cannot be directly meshed, the first automated task is to apply the chosen reconstruction method—either Blender's Voxel Remesh for low to medium densities or OpenVDB if also high-density specimens and specimens containing internal cavities are required to be simulated. All reconstruction parameters, such as voxel size or smoothing thresholds, are managed through a configuration Python file that can be adapted globally or for specific sample classes.

Once the geometry has been reconstructed into a watertight manifold, the workflow proceeds to the meshing stage handled by Gmsh. Automation here includes the import of STL files, automatic detection of closed volumes, and generation of 3D tetrahedral meshes. The characteristic length of mesh elements, which dictates mesh resolution, is set either globally or adaptively, depending on computational constraints and geometric complexity. The mesher's ability to handle millions of surface triangles without manual clean-up is a cornerstone of the workflow's scalability. If meshing fails due to memory limitations or subtle topological inconsistencies—issues occasionally observed during early pipeline development—the workflow logs the failure, flags the problematic specimen, and either retries with modified parameters or switches geometry reconstruction methods.

Once the volumetric mesh is successfully generated, boundary conditions and material properties are applied automatically. The gripping regions, always located at predefined and consistent positions across all specimens, are identified programmatically based on spatial coordinates. The left grip is

assigned fully constrained conditions, while the right grip receives the prescribed tensile displacement. Material parameters are inserted from a shared database, ensuring consistency across all simulations. The entire simulation input deck is then assembled in ESPRESO's native format.

The simulation itself is executed within a controlled HPC environment, where the workflow assigns the job to a compute cluster, manages resource allocation, monitors runtime performance, and retrieves the results. Because linear elastic simulations typically scale well across multiple cores, the workflow benefits significantly from parallel execution, dramatically reducing turnaround time per specimen. Once the solver completes, the workflow extracts the resulting displacement and stress fields, converting them into visualization-friendly formats or numerical summaries suitable for post-processing.

A central goal of the automation strategy is to reduce the need for manual analysis during intermediate stages. Thus, the workflow includes error-handling routines that detect common failure modes—such as non-manifold surfaces, excessive mesh distortion, or solver divergence—and apply corrective actions. These corrections may involve modifying voxel resolutions, applying additional smoothing to reduce numerical noise, or adapting meshing parameters. By capturing such issues early and automatically, the workflow ensures that only near-final results require human review.

Beyond the immediate goal of processing the existing dataset, the automation framework lays the groundwork for future extensions. The pipeline can be adapted to incorporate nonlinear materials, alternative boundary conditions, topology optimization studies, or surrogate modelling via machine learning—transforming it into a comprehensive digital testing environment. Moreover, the consistent structure of the pipeline facilitates integration with experimental databases, enabling automated comparison of numerical and physical results, parameter identification, or sensitivity analyses.

In essence, the automation strategy transforms a highly complex sequence of tasks—each requiring sophisticated software tools and immense computational resources—into a reproducible, scalable system capable of supporting large-scale numerical studies of lattice-filled additive-manufactured components. It bridges the gap between experimental and computational mechanics, enabling detailed exploration of geometry–mechanics relationships across a vast design space with minimal manual effort.

8. Conclusion

The work presented in this white paper demonstrates the feasibility, robustness, and practical value of a fully automated numerical workflow for simulating tensile tests of 3D-printed specimens with

complex internal lattice structures. Motivated by an extensive experimental program in which researchers investigated the mechanical influence of lattice topology and infill density on PLA tensile specimens, this study set out to replicate those results numerically and to develop a scalable digital framework capable of processing hundreds of specimen variants. The resulting workflow integrates advanced geometry reconstruction, high-resolution mesh generation, and massively parallel finite element analysis into a cohesive pipeline, supported entirely by open-source technologies and high-performance computing resources.

A central conclusion of this work is that accurate numerical simulation of additively manufactured lattice structures fundamentally depends on the ability to reconstruct watertight, topologically correct geometries from highly fragmented slicer-generated meshes. Traditional CAD repair strategies, including scripted workflows in Ansys SpaceClaim, were ultimately insufficient due to their inability to preserve internal voids or thin gaps between the lattice struts. By contrast, voxelization-based reconstruction—first through Blender's Voxel Remesh and, where necessary, through the more sophisticated OpenVDB library—proved indispensable. Blender's strengths lie in ease of automation and applicability to low and medium infill densities, while OpenVDB's sparse data structures and higher-fidelity isosurface extraction make it the only viable tool for reconstructing dense lattices, especially the Cubic and Adaptive Cubic ones, which contain internal voids.

The mesh generation stage, implemented through Gmsh, revealed both the power and the limitations of open-source meshing technologies when confronted with voxel-derived geometries of extreme complexity. Although Gmsh successfully generated high-quality tetrahedral meshes across the full range of specimens, the computational requirements—driven by surface meshes containing tens of millions of triangles—necessitated the use of HPC environments for both meshing and simulation. This finding underscores a broader insight: the structural richness of high-density lattice architectures is accompanied by unavoidable computational costs, not only in simulation but in every preceding stage of the workflow. Nevertheless, with appropriate memory resources and automated quality-control routines, the meshing process proved repeatable and scalable across the entire dataset.

The simulations themselves, conducted using the massively parallel ESPRESO solver, further demonstrated that linear elastic analysis provides a powerful and computationally efficient means of comparing lattice geometries, characterizing stiffness, and identifying fundamental stress-transfer mechanisms. While linear elasticity does not capture post-yield phenomena or failure patterns, it reliably reproduces early-stage deformation behaviour, yielding displacement and stress fields that align well with experimental observations and with physical intuition derived from lattice mechanics. The ability to visualize stress and deformation inside the specimen—particularly in internal regions inaccessible during physical testing—offers valuable insight into how lattice architecture influences mechanical performance. These insights can guide the design of optimized lattice structures, inform experimental interpretation, and support the development of predictive models in design automation and generative engineering.

From a methodological standpoint, the workflow succeeds not only because it solves the technical challenges of geometry reconstruction, meshing, and simulation, but also because it unifies these stages into an automated process. This level of automation is crucial for large-scale studies: it eliminates manual intervention, ensures systematic reproducibility, and enables efficient exploration of design spaces that far exceed the capacity of traditional experimental approaches. As additive manufacturing continues to push towards increasingly intricate lattice designs, the importance of such automated, high-throughput simulation workflows will only grow.

Despite its strengths, the present framework also highlights important limitations and opportunities for future development. The reliance on linear elastic material models restricts the predictive capability of the simulations to the pre-yield regime, leaving questions of ultimate tensile strength, strain localization, and fracture behaviour for future work. Incorporating nonlinear materials, damage models, or explicit fracture mechanics would allow for a more complete numerical reproduction of experimental results, albeit at significantly increased computational cost. Similarly, while the current meshing pipeline handles extremely large models, it remains sensitive to input resolution and may benefit from adaptive meshing techniques or multiscale modelling approaches that mitigate the need for uniformly fine volumetric meshes. Finally, the workflow is well suited for future integration with machine learning methods, potentially enabling surrogate models that predict mechanical response directly from lattice geometry without requiring full-scale finite element analysis.

In conclusion, the results of this study demonstrate that high-fidelity numerical simulation of lattice-filled 3D-printed specimens is not only feasible but can be performed at scale with appropriately designed tools and computational resources. By combining voxel-based geometry reconstruction, automated meshing, and parallel finite element simulation, the workflow provides a powerful digital counterpart to experimental testing. It paves the way for more sophisticated design studies, supports the validation of new lattice architectures, and establishes a foundation for computationally guided additive manufacturing research. The ability to process hundreds of geometrically complex specimens with minimal manual intervention represents a significant advancement in the numerical analysis of additively manufactured structures and opens the door to new engineering applications where the interplay between geometry and mechanics is both subtle and profound.

References

[1] https://docs.blender.org/

[2] https://www.openvdb.org/documentation/

[3] https://gmsh.info/

[4] https://www.it4i.cz/en/research/research-flagships/espreso-massively-parallel-library-for-engineering-applications

[5] https://www.prusa3d.com/page/prusaslicer_424/

[6] https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v242/en/spaceclaim/Discovery/user_manual/intro/t_intro.html

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, the Netherlands, Belgium, Luxembourg, Slovakia, Norway, Turkey, Republic of North Macedonia, Iceland, Montenegro, and Serbia. This project has received funding from the Ministry of Education, Youth and Sports of the Czech Republic.

VSB-TUO, IT4Innovations National Supercomputing Center, 17. listopadu 2172/15**,** 708 00 Ostrava-Poruba**,** Czech Republic