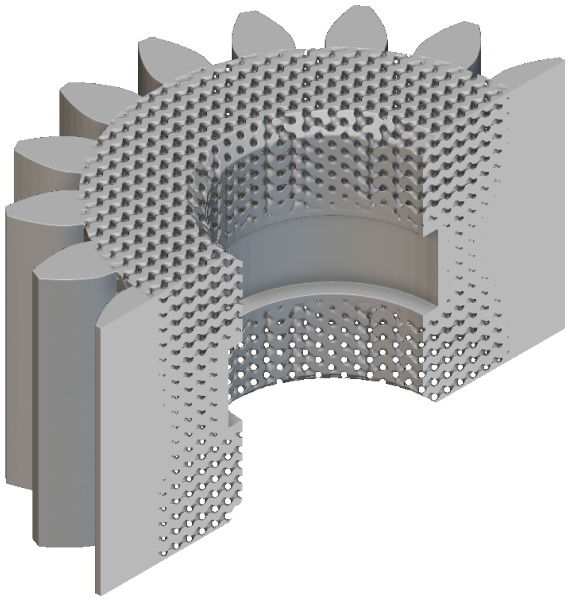
Additive Manufacturing and Topology Optimisation in Gearbox Design



Edoardo Rusconi

Supervisor: Prof. Ken Mao

School of Engineering
University of Warwick

This technical report is submitted for the degree of
BEng Mechanical Engineering

March 2025

ABSTRACT

The technical report investigates novel topology and infill optimisation methods to enhance the performance of knee powered orthosis with main objectives of weight minimisation for enhanced user comfort and gear stiffness maximisation to prevent backlash. The research embodies a comprehensive finite element method based framework, combining Matlab scripts and Abaqus simulations for the mechanical characterisation of complex gyroid lattice structures. 64 automated finite element simulations were executed, generating large amounts of data, over 960 figures per re-run, to develop regression models for Young's modulus across a variety of infill densities.

As such the technical report also posed a significant data visualisation challenge to concisely convey results using custom Matlab plots. These results are compared with nTopology software-based homogenisation method, which provided an orthotropic equivalent stiffness matrix from a single cell model, thereby validating the effectiveness of the iterative MATLAB approach. Despite minor limitations such as boundary discontinuities and mesh distorted elements, the automated process of creating 64 different simulations is novel in it's own right and the data acquired was demonstrated to be useful for further works and research.

While a homogenisation approach is not new and neither is the application of it to triply periodic minimal surface structure, the variation of level-set to determine homogenised properties at different infills has not been done before. It was found that the specific stiffness is highest at a level-set of 0 but the use of other densities applied to topology optimisation was deemed relevant.

The technical report's optimised structures achieved weight savings of up to 25.2% compared to the original geometry, showing excellent potential in the design of lighter and more comfortable powered orthosis systems.

In addition to the technical advances in lattice and infill optimisation, the thesis situates its contributions within the broader field of rehabilitation robotics. Since the most prevalent disability is mobility, backed up by statistical research from both UK and global perspectives, the work underscores the clinical and practical importance of reducing device weight and inertia. The integration of open-source actuator technologies, as embodied in the embrace of the OpenTorque model, also validates the potential for cost-effective and high-performance applications in knee orthosis.

Overall, the study provides a basis for the characterisation and optimisation of variable infill structures, bridging the gap between computational simulation and user-focused, practical design in rehabilitation robotics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Acronyms / Abbreviations

CAD	Computer Aided Design
DOF	Degrees of Freedom
EMG	Electromyography
FDM	Fused Deposition Modeling
FEM	Finite Element Method
KO	Knee Orthosis
MAD	Median Absolute Deviation
PA	Polyamide
PLC	Piecewise Linear Complex
PO	Powered Orthosis
SIMP	Solid Isotropic Material with Penalisation
SLS	Selective Laser Sintering
TO	Topology Optimisation
TPMS	Triply Periodic Minimal Surface
UTS	Ultimate Tensile Strength

1. INTRODUCTION

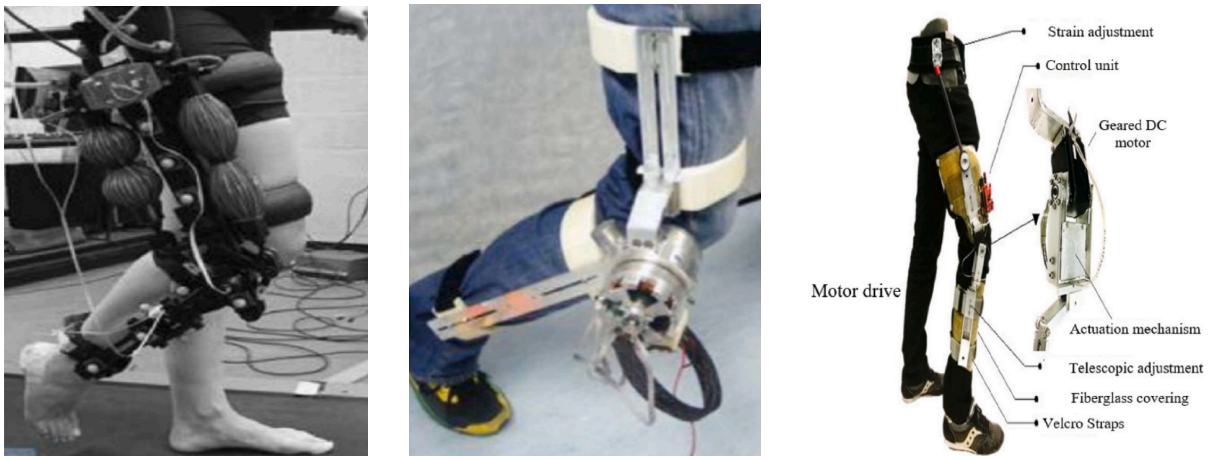
1.1 Background on Knee Injuries and Rehabilitation

1.1.1 Mobility, the most common disability

Traditional exoskeletons, known as Powered Orthosis (PO), aid individuals with little to no voluntary movement (Farris et al., 2011; Murray et al., 2015). However, a larger population would benefit from partial musculature assistance (Zhu et al., 2021). According to the UK Government, mobility is the most common type of impairment affecting 48% of the 16.1 million people identifying as disabled in the UK (Kirk-Wade et al., 2024). The current worldwide estimates are 527.8 million with a forecast to 2030 that projects an increase (Chen et al., 2023). The most common types of mobility aids are rollators, walking frames, canes, crutches, wheelchairs and mobility scooters (Jung and Ludden, 2019).

1.1.2 Use of Powered Orthosis

Knee PO has emerged as a promising technology with significant potential for enhancing the rehabilitation process and improving the quality of life for individuals with knee-related impairments (Dereshgi et al., 2023). Further improvements of quasi-passive PO should focus on materials, actuators, controllers, energy efficient, safety, and cost effectiveness of the exoskeletons (Al-Hayali et al., 2021).



(a) (Beyl et al., 2011) (b) (Karavas et al., 2013) (c) (Claros et al., 2016)

Fig. 1.1 3D printed powered knee orthosis (Trivedi and Joshi, 2024)

1.1.3 Rehabilitation Robotics

The field of lower limb rehabilitation robotics is broadly comprised of therapeutic and assistive robots, covering a range of different forms of post-traumatic, post-operative and elderly health care services where direct physical interaction with a robot system can enhance the patient recovery or act as a replacement for the lost functionality (Bhardwaj et al., 2021).

1.2 Research Justification

For the widespread adoption of knee PO, Mannion et al. (2024) identified user comfort as a key limiting factor. A decrease in weight would through topology optimisation would improve user comfort. Barrera Sánchez et al. (2022) found an increased Electromyography (EMG) for an actively powered Knee Orthosis (KO) and concluded that due to excess weight, inertia and friction was introduced. Once again highlighting the benefit of topology optimisation. Weight reduction can be achieved by minimising strain energy in Abaqus software to a set volume. For gears to reduce backlash, later explored in Section 2.2.5.2, stiffness needs to be maximised. A literature gap is also subsequently justified both academically and commercially (Section 2.1) through a patent analysis

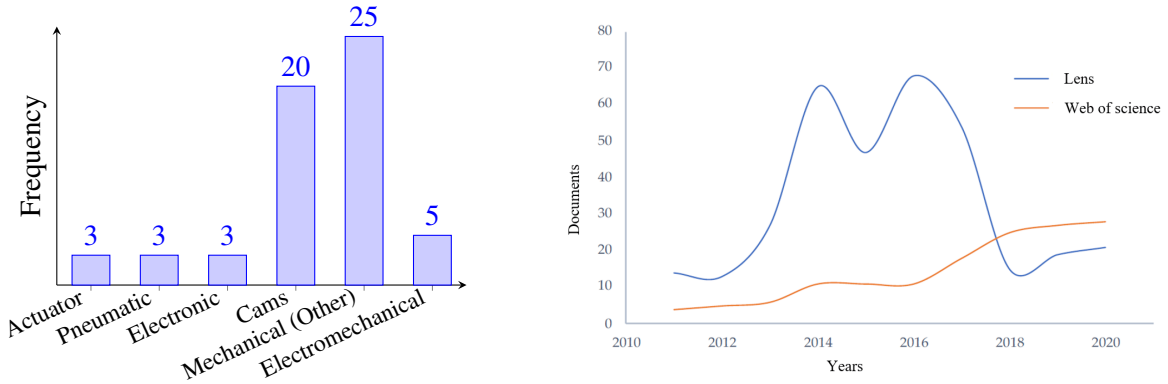
1.3 Objectives

1. Topology and infill optimisation
    (a) Increase user comfort through weight reduction (Section 1.1.2).
    (b) Reduce gear backlash by maximising gear stiffness and back-drivability (Section 1.2).
2. Infill density characterisation and homogenisation
    (a) Use an iterative Finite Element Method (FEM) approach to estimate infill lattice properties.
    (b) Validate chosen infill using Topology Optimisation (TO) and data from Objective 2a.
3. Adapt OpenTorque open-source quasi-direct drive into an optimised knee PO (Levine, 2019).
    (a) Select and validate gear geometry and gear ratio of 5:1 based on OpenTorque model.
    (b) Manufacturing cost of below £100, exceptionally low for a medical device.

2. Literature Review

2.1 Patent Analysis

In Figure 2.1 a patent search identifies a distinct literature gap for actuator based KO, with a study from Meda-Gutiérrez et al. (2021) having found only 3 patents in this subject area, of which none include 3D printable components. The most common KO braces available on the market use purely mechanical system of which a large portion are exclusively cam based.



(a) Sorted by technology

(b) KO filings per year

Fig. 2.1 Patent analysis distilled from 8618 search results (Meda-Gutiérrez et al., 2021)

2.2 Single Stage Gear Systems for Robotic Actuators

2.2.1 Cycloidal Drive

A cycloidal drive consists of a high-speed shaft with an eccentric bearing. The cycloidal discs rotate eccentrically, engaging with the fixed outer pins on the disc lobes. Driving pins with rollers transmit motion from the cycloidal discs to the output shaft. The mechanical system is illustrated in Figure 2.2.

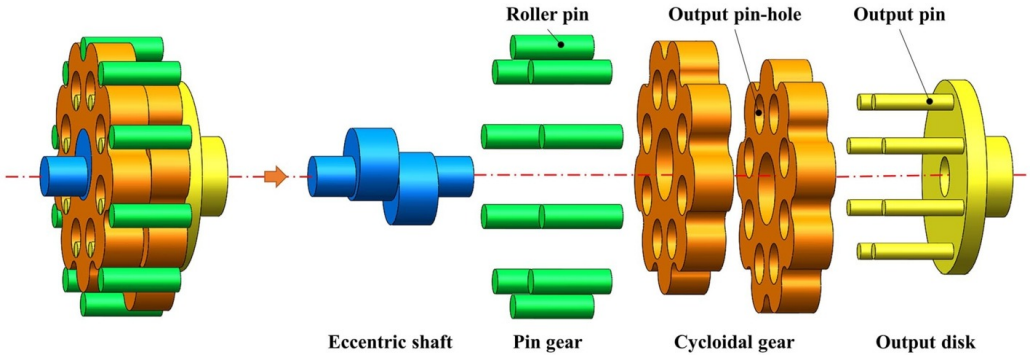


Fig. 2.2 Cycloidal gearbox (Qi et al., 2024)

As derived by Soham et al. (2024) the gear ratio (N) in a cycloidal gearbox is given by Equation 2.1.

$$N = \frac{N_r}{N_c - N_r} \tag{2.1}$$

Where N_r is the number of rollers (pins) on the outer ring, and N_c is the number of lobes on the cycloidal disc. A single stage cycloidal gear induces an imbalance in gear inertia and undesired vibrations. To mitigate this a counterbalancing disk is incorporated (Zhu et al., 2024). An offset two stage cycloidal gearbox would be balanced but conflicts with Objective 3a and the alternative is too heavy for Objective 1a.

2.2.2 Harmonic Drive

A harmonic drive, also known as strain wave gearing, illustrated in Figure 2.3 is composed of a cam with flexible bearing functioning as an elliptical wave generator, a flexspline, circular spline and a cross roller bearing. Its very high gear ratio is unsuitable for the 5:1 reduction ratio outlined in Objective 3a.

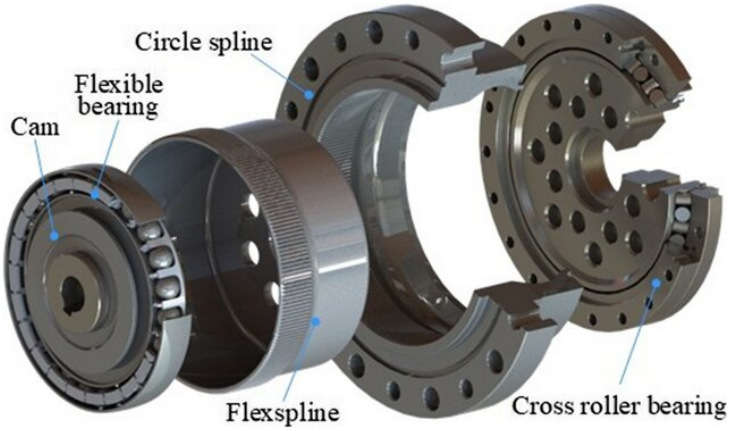


Fig. 2.3 Harmonic Drive Design (Xiao et al., 2025)

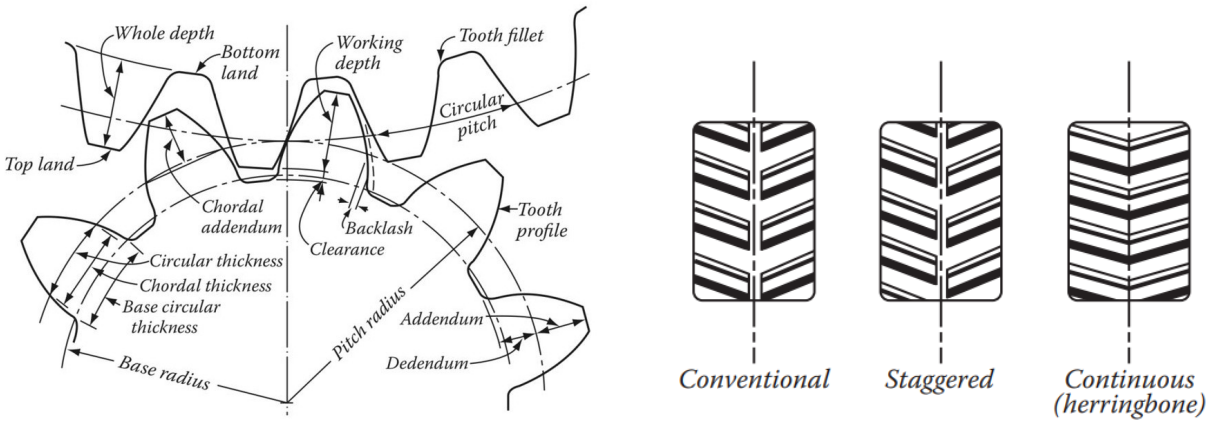
The harmonic drive was introduced by (Musser, 1959) and filed as a US patent, the gear ratio (N) is defined in Equation 2.2 and is dependent on the teeth ratio between circular spline and flexspline.

$$N \approx \frac{N_{\text{circular spline}} - N_{\text{flexspline}}}{N_{\text{flexspline}}} \tag{2.2}$$

2.2.3 Planetary Gearbox

2.2.4 Gear Geometries

Figure 2.4a details several critical dimensions of spur gears, backlash is labelled and is an intentional clearance to prevent binding. Spur gears have no axial loads, this is true also for double-helical gear geometry in Figure 2.4b which are self centering but introduce complex teeth loads.



(a) Spur gear geometry (b) Full assembly

Fig. 2.4 Gear geometries (Dudley and Townsend, 1991)

2.2.4.1 Gear Validation - The Lewis Method

In Equation 2.3, σ represents the bending stress, while $F_{\text{effective}}$ is the effective tangential force per tooth. The denominator consists of the face width of the gear b, the module m, which defines the tooth size, and the Lewis form factor Y, which accounts for tooth shape and stress concentration. This method is proposed in Dudley's gear handbook (Dudley and Townsend, 1991) and assumes the gear teeth to be a cantilever beam and is useful for initial validation.

$$\sigma = \frac{F_{\text{effective}}}{b \cdot m \cdot Y}. \tag{2.3}$$

2.2.5 Gearbox Suitability for Knee Powered Orthosis

Due to the constraints from Objective 3a harmonic and cycloidal drives have been deemed unsuitable for the required knee PO application. While a cycloidal gearbox with a gear ratio of 11:1 has been proposed by Roozing and Roozing (2022) it would be impractical for a lower gear ratio, such as 5:1 proposed in Objective 3a.

2.2.5.1 Torque Density and Reflected Inertia

Table 2.1 shows the scaling laws for the torque density of several gear mechanisms.

Table 2.1 Scaling laws for maximum continuous output torque and reflected inertia based on diameter (d), length (L), transmission ratio (i) and number of stages (a) (Saerens et al., 2019)

Drive Mechanism	**Max Output Torque**	**Reflected Inertia**	**Suitability**
Parallel Shaft Gear Train (PSGT)	$\frac{Ld^2}{a}$	$\frac{Ld^4i^2}{a}$	Moderate torque, high inertia. Suitable for slower joints but not ideal for dynamic legged motion.
Planetary Gear Train (PGT)	$\frac{Ld^2}{a}$	$\frac{Ld^4i^2}{a}$	Compact and efficient, but high inertia limits dynamic performance for high gear ratios.
Harmonic Drive	d^3	Ld^4i^2	High torque density and moderate inertia. Ideal for precise and dynamic movements, suitable for knee joints.
Cycloidal Drive	$\frac{d^4}{L}$	Ld^4i^2	High torque and low backlash, ideal for dynamic legged motion and knee actuators.
Ball Screw	d^3	Ld^4	Linear actuators not suitable for knee PO.

2.2.5.2 Backlash Hysterisis

Backlash hysterisis occurs when backdriving and is the combination of backlash, play between gears that cause positional uncertainty and hysterisis, the lag and energy loss due to friction and deformation,

Some backlash is inevitable for proper tooth meshing and engagement. The Preisach model and the Prandtl–Ishlinskii model are two well-known models to simulate backlash hysteresis (He et al., 2017). Simulations of these models, as discussed in Gu et al. (2014), show a reduction of rotational position uncertainty by around 50% to 70%.

Harmonic drives suffer from fatigue-induced flexspline wear, making them particularly susceptible to increased hysterisis, as seen in Figure 2.5. This would be exacerbated in a 3D printed harmonic drive making the gearbox type unsuitable for knee PO.

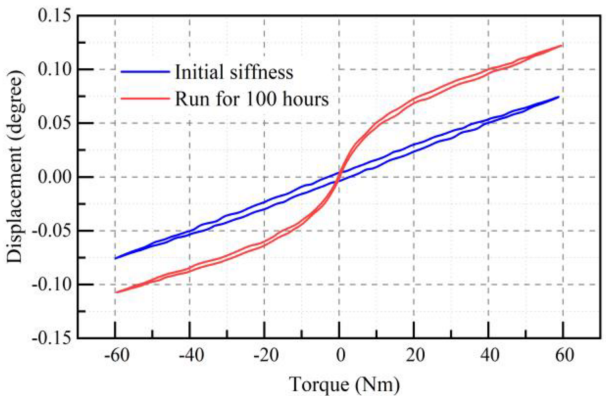
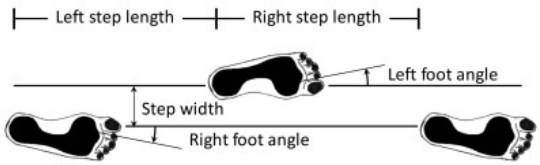


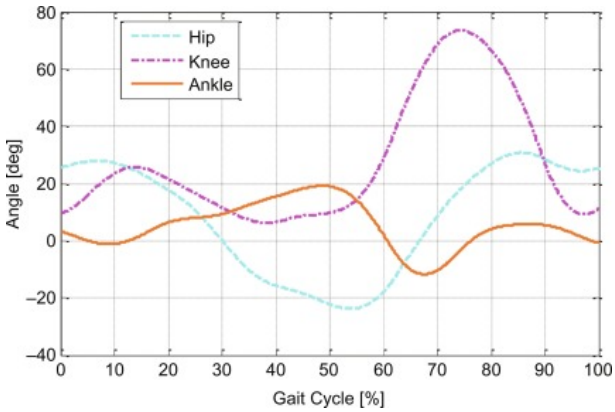
Fig. 2.5 Hysterisis in a harmonic drive (Tang et al., 2021)

2.3 Gait Cycle

The Gait cycle, experimentally analysed in Figure 2.6 is the movement that describes bi-pedal motion and is divided into the stance and swing phases. Laribi and Zeghloul (2020) used motion captured data, the standard for Gait cycle analysis, due to no inertial sensor interference, the data highly reliable (Homes et al., 2023). However, lack of error bars and an aggregate average of cycles is a limitation of this study.



(a) Gait phases

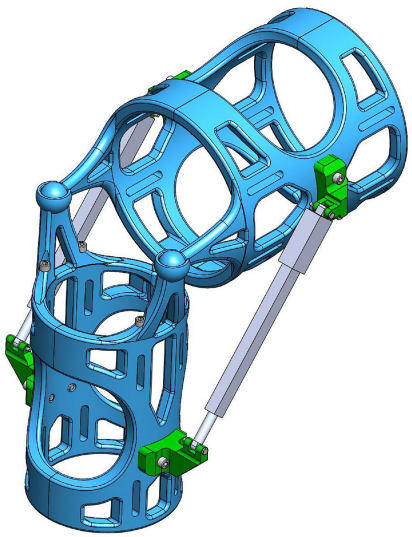


(b) Gait cycle

Fig. 2.6 Experimental Gait cycle using Vicon Nexus motion capture system (Laribi and Zeghloul, 2020)

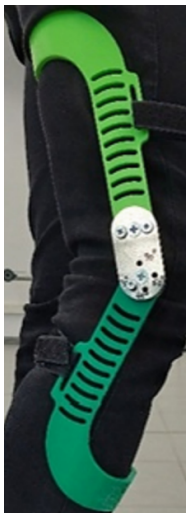
2.4 3D Printed Orthotic Knee Braces

Figure 2.7 shows three different 3D printed passive knee brace designs. Passive knee braces function using a linear or torsional spring. By storing the potential energy during knee flexion an assistive moment is applied for knee extension, assisting in the swing phase of the Gait cycle (Section 2.3) (Budarick et al., 2019). An active knee brace PO assists the entire Gait cycle making it suitable for more applications.



(a) Boolos et al. (2022)

(b) Hendricks et al. (2018)

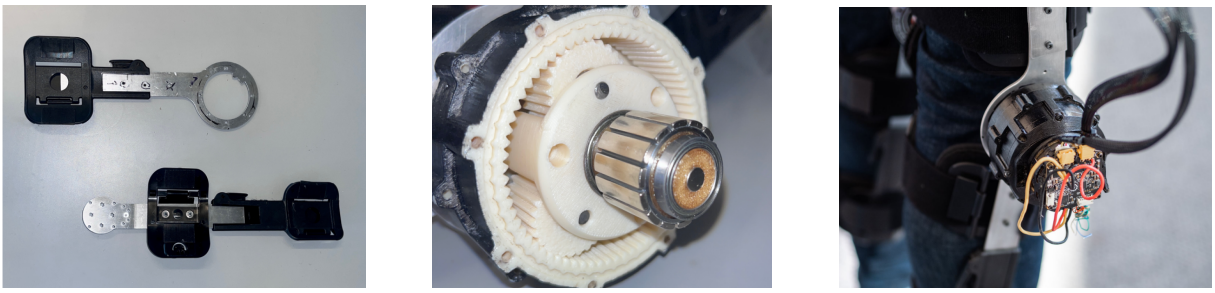


(c) Torcătoru and Tulică (2024)

Fig. 2.7 3D printed passive knee braces

2.5 3D Printed Powered Knee Orthosis

Figure 2.8 illustrates a powered KO from Sanchez et al. (2023). Figure 2.8a is a pair of knee braces made from 7075 aluminium. Figure 2.8b, the planetary gearbox is printed using ULTEM 1010 resin filament for Fused Deposition Modeling (FDM) printing and ASA thermoplastic. Figure 2.8c is the full mounted powered KO in use.



(a) Braces (b) Actuator (c) Full Assembly

Fig. 2.8 3D printed powered KO (Sanchez et al., 2023)

2.6 Triply Periodic Minimal Surface Infill

A generic Triply Periodic Minimal Surface (TPMS) can be defined using Equations 2.4, 2.5 and 2.6 as presented by Chris-Amadin and Ibhadode (2024) where and c, c_1, c_2 are iso-value constants.

$$f(x,y,z) = c \tag{2.4}$$

$$f(x,y,z) \leq c \ \vee \ f(x,y,z) \geq c \tag{2.5}$$

$$c_1 \leq f(x,y,z) \leq c_2 \tag{2.6}$$

Equation 2.4 is known as the level-set and defines a minimal surface in the TPMS. Equation 2.6 is the bounded volume inequality and represents a volume fraction from the minimal surface. Infill percentage can be varied by changing the level-set c or the bounded volume inequality. For a gyroid structure, the function $f(x,y,z)$ in Equations 2.4, 2.5 and 2.6 is given by Equation 2.7.

$$\sin(x)\cos(y) + \sin(y)\cos(z) + \sin(z)\cos(x) = f(x,y,z) = c \tag{2.7}$$

Ashby and Gibson (2003) provided useful results relating the characteristics of cellular solids.

$$E^* = C_1 \rho^{*n}, \tag{2.8}$$

Where the scaling law for the elastic modulus E^* is given by:

$$E^* = \frac{E_{\text{latt.}}}{E_{\text{sol.}}}, \tag{2.9}$$

and $E_{\text{latt.}}$ and $E_{\text{sol.}}$ are the elastic moduli of the lattice structure and the constituent material, respectively.Similarly,

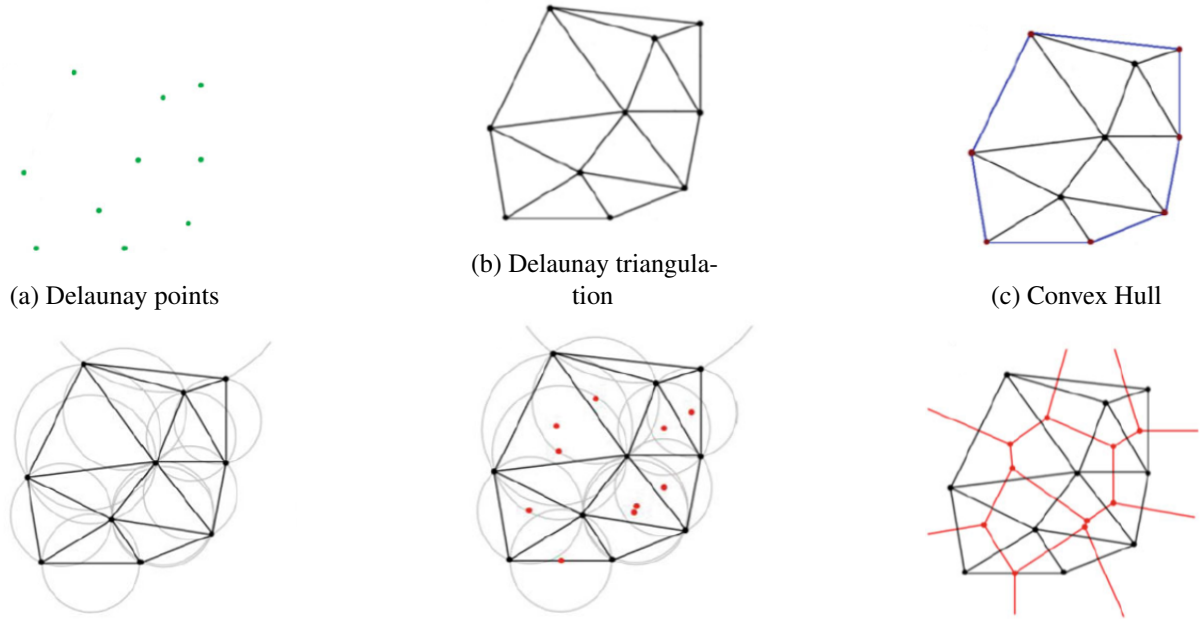
$$\rho^* = \frac{\rho_{\text{latt.}}}{\rho_{\text{sol.}}}, \tag{2.10}$$

where $\rho_{\text{latt.}}$ and $\rho_{\text{sol.}}$ are the densities of the lattice structure and the constituent materials, respectively.

ρ^* is referred to as relative density or volume fraction; and can take values from 0 to 1, where 1 represents a fully solid structure. The prefactor C_1 in Equation 2.8 was given by Gibson and Ashby.

2.6.1 Gibbon Matlab Library

2.6.1.1 TetGen



(a) Delaunay points

(b) Delaunay triangulation

(c) Convex Hull

(d) Delaunay circumcircles

(e) Delaunay circumcenters

(f) Voronoi diagram

Fig. 2.9 Delaunay triangulation and Voronoi for mesh generation (Dinas and Martínez, 2020)

The Delaunay triangulation method ensures maximised minimum angles and avoiding poorly formed mesh elements, critical for numerical stability. This is provided through the TetGen C++ code discussed in Si (2015) compiled to Matlab Executables (.MEX). Thus, allowing near-native C++ code execution for computationally intensive TetGen code that bypasses Matlab's interpreted code execution, running directly on the CPU. Figure 2.10 shows the Piecewise Linear Complex (PLC) mesh generation process and Delaunay method. TetGen also supports second-order elements through the '-o2' option flag.

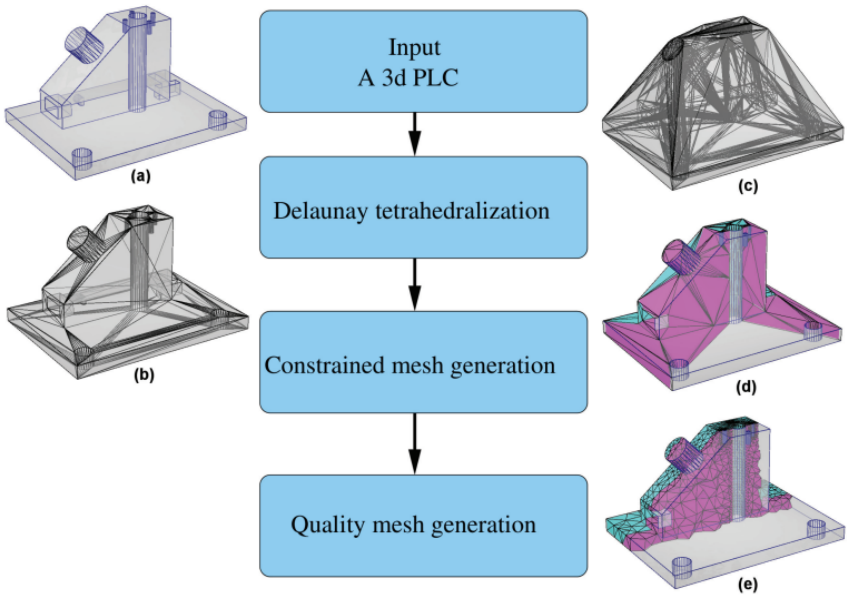
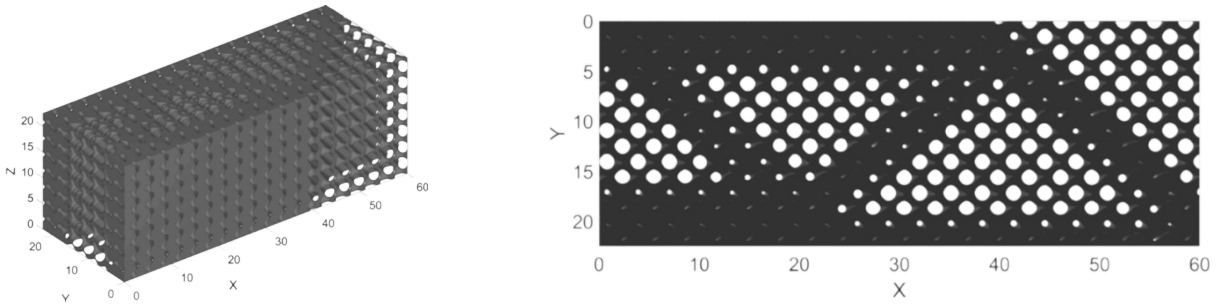


Fig. 2.10 a) PLC 3D Geometry, b) PLC Boundaries, c) Delaunay tetrahedralization, d) Geometric constraints, e) Refine tetrahedral mesh (Si, 2015)

The PLC, first introduced by (Miller et al., 1999) is the mesh, composed of pieced together edges and faces. Unlike parametric geometry such as a .STEP file, it is discretised into edges and faces.

2.6.2 LatticeWorks Matlab Library

LatticeWorks is an integrated framework for the automated design, optimisation, and fabrication of lattice structures. It encompasses various lattice generation methods, including single- and multi-morphology lattices, linearly graded lattices, different boundary shapes, and cell arrangements (Vafaeefar et al., 2025). Figure 2.11 shows a density weighted TPMS gyroid lattice of an infill topology optimised beam.



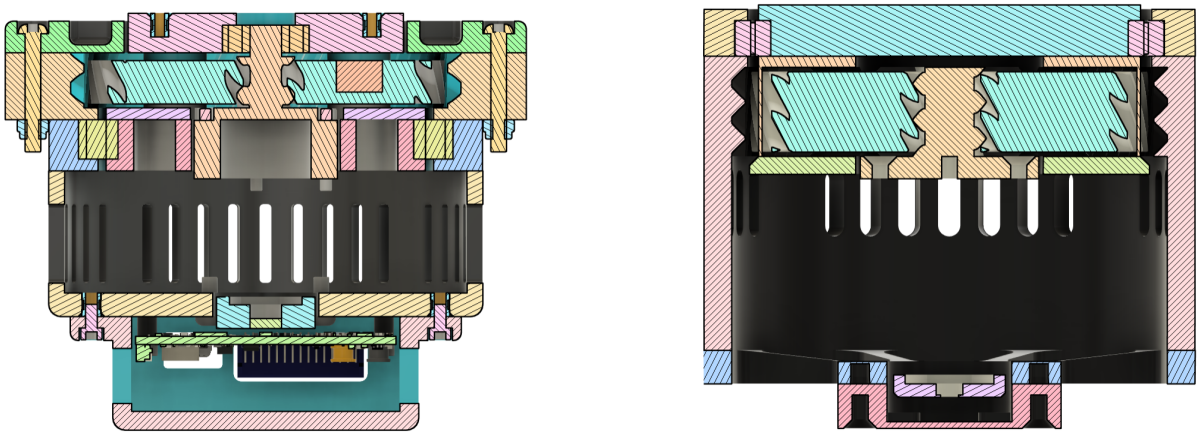
(a) (b)

Fig. 2.11 Topologically optimised variable infill TPMS gyroid (Vafaeefar et al., 2025)

2.7 OpenQDD and OpenTorque Comparison

Figure 2.12 illustrates the open source 3D printed actuators by Musa (2023) and Levine (2019), neither of these was specifically intended for knee PO but were instead designed as legged robotic actuators, heavily inspired to be the 3D printed counterparts to the MIT Cheetah Bledt et al. (2018).

OpenQDD has not been academically reviewed. However, OpenTorque has been extensively tested with Yoshida et al. (2023) concluding a rated continuous torque of 37.6 Nm and a maximum torque of 80 Nm.



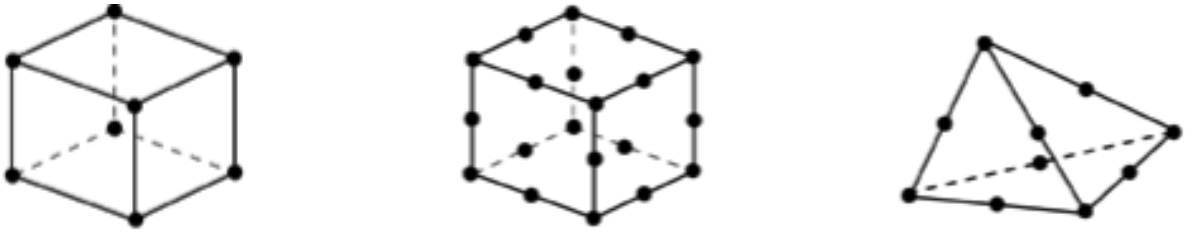
(a) OpenQDD (Musa, 2023) (b) OpenTorque (Levine, 2019)

Fig. 2.12 3D printed actuator, literature comparison

The gears for both designs are shown in blue. Further comparison on their specific dimensions to the design specifications later proposed are discussed in Table 3.2.

2.8 Finite Element Methods (FEM)

2.8.1 Abaqus Element Types



(a) C3D8 (b) C3D20 (c) C3D10

Fig. 2.13 Abaqus Elements (Systèmes, 2016)

Figure 2.13a and Figure 2.13b are Hexagonal (Hex) elements of first-order and second-order respectively and Figure 2.13c is a second-order Tetrahedral (Tet) element. Each node as these are continuum element

contains 3 Degrees of Freedom (DOF), in the U_x, U_y, U_z directions which are known as U1, U2, U3 in Abaqus. This constitutes 24 DOF's, 60 DOF's and 30 DOF's for CSD8, C3D20 and C3D10 respectively.

As discussed in Section 2.6.1.1, tetrahedral elements provide the most robust meshing for complex geometry, essential for automated mesh generation without requiring manual partitioning to allow Hex meshing. While the reduced DOF's per element makes tetrahedral elements less computationally efficient using them is necessary for automated mesh generation, or complex geometry.

2.9 Topology Optimisation

The first Solid Isotropic Material with Penalisation (SIMP) TO in Matlab was done by Sigmund (2001), later improved by Andreassen et al. (2011) and then revised by Ferrari and Sigmund (2020).

2.9.1 Defining Design Domain

A parametric solid is meshed into tetrahedral or quadratic elements as illustrated in Figure 2.13 using a meshing algorithm such as TetGen (Section 2.6.1.1). Penalisation encourages to parametrise material density to 1 = solid and 0 = void (Siva Rama Krishna et al., 2017).

Bendsøe and Sigmund (2004) emphasises that the discretisation resolution directly impacts solution accuracy and computational cost, requiring a balance between mesh refinement and practicality.

The importance of filter size in proportion to the design domain is evident in Figure 2.14 where the number of voxel elements of a 2D topology optimisation result is post-processed by the same size filter radius. As suggested by Panesar et al. (2017), a lower SIMP penalisation or a larger filter size can produce sparser densities, which are ideal for a graded TPMS gyroid lattice.

(a) Unfiltered 60×20 elements (b) Filtered (a) (c) Unfiltered 40×40 elements (d) Filtered (c)

Fig. 2.14 Effect of filter radius $R = 1.5 \times 10^{-2}$ on different domain sizes (Bourdin, 2001)

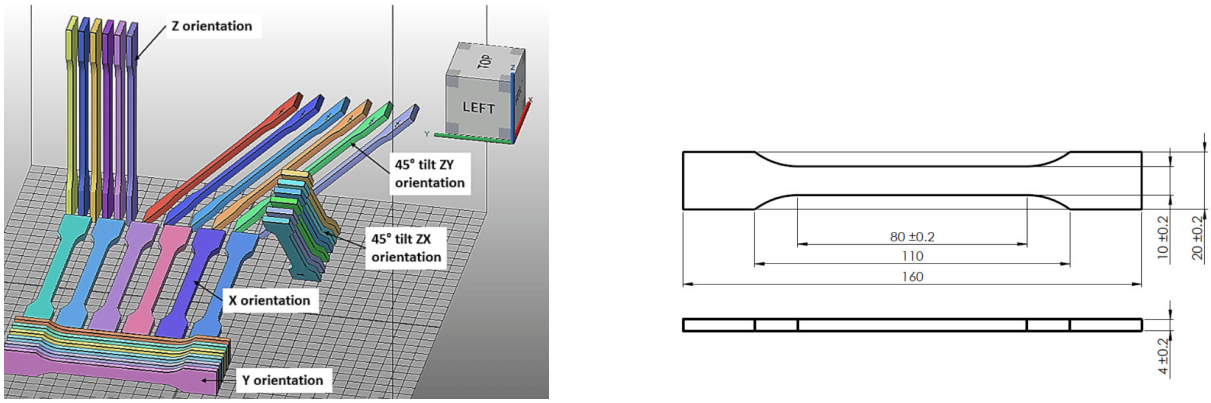
2.9.1.1 Material Definition

Polyamide (PA) 12, also known as nylon, is commonly used in Selective Laser Sintering (SLS) and FDM 3D printing. A summary of SLS, PA 12 properties is collated in Table 2.2.

Table 2.2 PA 12 properties of SLS 3D printing based on print direction (Rosso et al., 2020)

E [GPa]	UTS [MPa]	ε_{max} %	Printing direction	Reference
1.76 ± 0.02	43.6 ± 0.5	31.6 ± 2.9	XY plane	(Xu et al., 2019)
1.68 ± 0.04	47.6 ± 1.4	6.6 ± 0.5	X axis	(Sillani et al., 2019)
1.61 ± 0.06	40.6 ± 3.2	3.7 ± 0.5	Z axis	(Cai et al., 2021)
1.72 ± 0.01	45.1 ± 0.5	10 ± 0.1	XY plane	(Salazar et al., 2014)
1.64 ± 0.01	46.4 ± 0.1	16.9 ± 0.1	X axis	(Pilipović et al., 2018)
2.16 ± 0.05	49 ± 1.7	4 ± 0.3	Z axis	(Pilipović et al., 2018)
1.39 ± 0.03	44 ± 0.1	27.6 ± 2.6	X axis	(Van Hooreweder and Kruth, 2014)
1.61 ± 0.10	43.9 ± 0.7	26.6 ± 2.9	Y axis	(Cai et al., 2021)
1.22 ± 0.03	39.6 ± 0.2	14.7 ± 1.1	Z axis	(Rosso et al., 2020)
1.87 ± 0.04	46.9 ± 0.9	10.3 ± 1.9	Z axis	(Rosso et al., 2020)

Figure 2.15 illustrates the definition of the 3D print directions from Rosso et al. (2020) and consistent with Table 2.2. The tensile samples tested, sketched in Figure 2.15b are consistent with the ISO 527 standard for the mechanical properties of plastics ISO (2012), often used in 3D printed samples.



(a) C3D8 (b) ISO 527 Sample

Fig. 2.15 3D Printed Samples (Zakręcki et al., 2024)

Some limitations of the sources include the lack of standardised processing parameters. The material properties and quality of the final SLS print is highly dependent on processing parameters, particularly laser power and speed (Jabri et al., 2023; Ligon et al., 2017; Radu et al., 2025; Singh et al., 2020). The difficulty of such parametric study is shown by Zhang et al. (2020) where a few input parameters resulted in 5,940,000 possible combinations, requiring neural networks to solve for multi-parameter regression as it is computationally cheaper than standard numerical methods (Kshirsagar and Rathod, 2012; Vogl

et al., 1988). Infill percentage has the most influence on the performance of a 3D printed gear (Zhang et al., 2020).

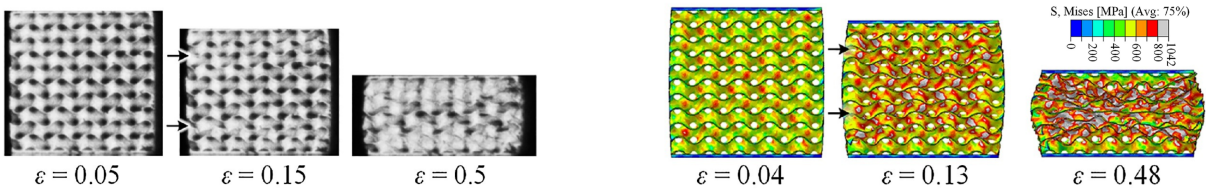
2.9.2 2.5 Dimensional TO

In Figure 2.16 a 2.5D approach is used. Penalisation encourages to parametrise material density to 1 = solid and 0 = void (Siva Rama Krishna et al., 2017). A low penalisation factor of 1 in Figure 2.16a yields non-convergence to 0 - 1 densities allowing for a smoother gradient. This has been identified as a useful feature by Panesar et al. (2017) for creating functionally graded lattices, yielding smoother stress transitions and precisely tailored mechanical properties.

(a) P=1 vertical (b) P=3 truss (c) P=3 variable (d) Microstructure

Fig. 2.16 TO using 2.5D approach results where (P) is penalisation used (Kandemir et al., 2018)

2.10 Failure Modes of TPMS Structures



(a) Gyroid compression test (b) Gyroid FEA stress distribution

Fig. 2.17 Gyroid metal structure failure, arrows indicate failed layers (Zhang et al., 2018)

2.11 Conclusion

In conclusion, the current state of knowledge on KO, including the mechanisms, materials, and design strategies employed in their fabrication, is reflected in this literature review. The patent review shows a clear trend towards the use of latest technologies, such as 3D printing and sophisticated gear systems, which are critical in enhancing the performance and functionality of powered KO.

The quest for alternative gear systems, including cycloidal drives, harmonic drives, and planetary gearboxes, highlights the importance of torque density and reflected inertia in selecting appropriate mechanisms for knee applications. The analysis shows that while harmonic and cycloidal drives have advantages, their weight and backlash limitations make them unsuitable for the intended applications in knee PO. On the other hand, the suitability of a planetary gearbox is evident for a gear ratio of 5:1.

Moreover, research into gait cycles and their relevance to orthotic design highlights the need for individualisation of devices to support natural movement. The technological advancement of 3D printing technologies, particularly with materials like PA 12, provides the means for customisation and optimisation of orthotic geometry, reflecting the potential for enhanced patient benefits.

This review points to the convergence of biomechanics, materials science, and mechanical engineering in the construction of KO to advance to new technology that may greatly improve mobility and quality of life for patients who are knee-impaired.

2.11.1 Additional Critical Review of Sources

A "2.5D" approach has been used by Kandemir et al. (2018); Yarlagadda et al. (2022) where 2D toplogy optimisation is performed then variable extruded. This approach however, assumes a uniform infill and fails to utilise the main advantages of additive manufacturing compared to traditional manufacturing processes: it's flexible manufacturing process with complex geometry and infills (Iftekar et al., 2023).

Recent studies such as Kim et al. (2021); Roozing and Roozing (2022); Yoshida et al. (2023) introduce 3D-printed gear actuators as promising robots in robotics technology, highlighting benefits such as rapid prototyping and high design flexibility. However, these works share the same shortcomings, lacking long-term durability testing and brief mechanical characterisation, particularly with regards to material anisotropy of polymers, insufficiently exploring the ideal infill optimisation or variable infill.

Zhang et al. (2018) provides an excellent, rigorously detailed analysis for metal structures; however, their findings are not fully transferable to other additive manufacturing processes such as SLS. In contrast, FDM exhibits a higher tendency for inter-layer failure due to visco-elastic and brittle behavior of polymer interfaces (Thumsorn et al., 2022).

Panesar et al. (2017) claims that a graded lattice structure was more resilient for the paper's proposed load case but did not conduct any physical or fatigue testing. The manufacturing concerns are very well addressed, for example considering the added waste of additional supports when removing material into the optimisation process. The proposed gyroid TPMS structure is self supporting (Sankineni and Yennam, 2021).

3. METHODOLOGY

3.1 Actuator Requirements

The range of motion, moment, speed and forces on the knee during walking and squatting are summarised in Table 3.1. The knee range of motion constraints coincide with the previous Gait cycle literature analysis in Figure 2.6b. To meet Objective 1a and user comfort the range of motion and maximum knee joint speed should not be limited. The use of a motor driven planetary gearbox eliminates range of motion constraints and the 8318 KV120 motor from Eaglepower (2024) can achieve 1203 RPM under minimal load which equates to 125.98 rad/s.

Table 3.1 Actuator requirements for squatting and walking (Neumann and Kelly, 2025; Shelburne et al., 2004; Tang et al., 2022)

Parameter	Walking	Squatting
Range of motion (deg)	15–60	0–150
Maximum knee joint moment (Nm)	40	60
Maximum knee joint speed (rad/s)	4.3	2.4
Maximum force on knee	x4.6 body weight	x7.8 body weight

3.2 Open QDD Design Analysis

Table 3.2 Comparison of actuator and gear specifications

Specification	**OpenTorque** (Levine, 2019)	**OpenQDD** (Musa, 2023)	**Proposed Actuator**
Number of planets	3	3	3
Gear thickness	18mm	11mm	18mm
Planet bearing ID	5mm	6mm	5mm
Planet bearing OD	16mm	28mm	16mm
Planet gear teeth	27	28	18
Sun gear teeth	9	8	12
Ring gear teeth	63	64	48
Carrier bearing ID	80mm	75mm	75mm
Carrier bearing OD	96mm	95mm	95mm
Sun bearing ID	–	8mm	–
Sun bearing OD	–	22mm	–
Carrier bearing thickness	8mm	–	10mm
Case ID	90.5mm	95mm	90.5mm
Case OD	110mm	110mm	110mm

3.2.1 Orthotropic Material

Using Table 2.2, where directions x, y, z are equivalent to 1, 2, 3. The elastic modulus (E) can be determined.

$$E_1 = 1.57 \pm 0.08 \text{ GPa} \qquad E_2 = 1.61 \pm 0.10 \text{ GPa} \qquad E_3 = 1.715 \pm 0.18 \text{ GPa}$$

Using Equation 3.1, the shear modulus (G) can be estimated using v, determined to be 0.38 in all directions (Giannopoulos, 2019).

$$G = \frac{E}{2(1+v)} \tag{3.1}$$

Table 3.3 Orthotropic elastic properties of SLS-printed PA 12

(a) Elastic moduli

Direction	E (GPa)
1	1.57 ± 0.08
2	1.61 ± 0.10
3	1.715 ± 0.18

(b) Poisson's ratios and shear moduli

Plane	v	G (GPa)
12	0.38	0.57 ± 0.03
13	0.38	0.58 ± 0.04
23	0.38	0.62 ± 0.07

Due to current limitations in the Gibbon library (Section 2.9.1.1) an isotropic model was used due to the relatively low variance in directional material properties observed in SLS 3D printing. From Table 3.3a, using an average of directional properties a Young's modulus E of 1.666 GPa $\pm$ 0.039 was calculated with a Poisson's ratio v of 0.38.

The gyroid TPMS geometry will therefore be the main cause of non-linearity. SLS avoids the layer-dependent weaknesses inherent in FDM, enabling the production of stronger gears. In contrast, FDM printed gears must be oriented with the layers lying flat, aligned with the weakest direction, to preserve tooth geometry and ensure that any layer-induced weaknesses are evenly distributed.

3.2.2 Abaqus Unit System

The SI (mm) unit system will be used, this is an important definition since Abaqus, similarly to most commercial FEA software is unitless. Table 3.4 shows the proposed modified SI (mm) system where kN is changed to N to be more intuitive. Changing the elastic modulus to 1666 MPa (N/mm^2).

Table 3.4 Comparison of proposed SI unit system

Quantity	**SI (mm) System**	**Modified SI (mm)**
Length	mm	mm
Pressure	GPa (kN/mm^2)	MPa (N/mm^2)
Time	s	s
Force	kN	N
Moment	kN·mm	N·mm
Area	mm^2	mm^2

3.2.3 Material Homogenisation

An orthotropic material is a type of anisotropic material where x', y', z' are the axis of material symmetry, meaning the perpendicular directions by which material properties are constant. The stiffness matrix can be fully defined with the Young's modulus E, Poisson's ratio v and shear modulus G as discussed by Zienkiewicz et al. (2013), this is generalised using Equation 3.3 in Voigt notation.

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} \tag{3.2}$$

The orthotropic material definition is given in Equation 3.3 (Zienkiewicz et al., 2013). The flexibility modulus is the inverse of the stiffness matrix C, such that $\sigma_i = C_{ij}\varepsilon_j$, and the flexibility matrix S is $S_{ij} = C_{ij}^{-1}$ (Kabe and Sako, 2020).

$$\begin{bmatrix} \varepsilon_{x'} \\ \varepsilon_{y'} \\ \varepsilon_{z'} \\ \gamma_{x'y'} \\ \gamma_{y'z'} \\ \gamma_{z'x'} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_{x'}} & -\frac{v_{x'y'}}{E_{y'}} & -\frac{v_{x'z'}}{E_{z'}} & 0 & 0 & 0 \\ -\frac{v_{y'x'}}{E_{x'}} & \frac{1}{E_{y'}} & -\frac{v_{y'z'}}{E_{z'}} & 0 & 0 & 0 \\ -\frac{v_{z'x'}}{E_{x'}} & -\frac{v_{z'y'}}{E_{y'}} & \frac{1}{E_{z'}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{x'y'}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{y'z'}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{z'x'}} \end{bmatrix} \begin{bmatrix} \sigma_{x'} \\ \sigma_{y'} \\ \sigma_{z'} \\ \tau_{x'y'} \\ \tau_{y'z'} \\ \tau_{z'x'} \end{bmatrix} \tag{3.3}$$

3.3 Gear Geometry

This section is necessary to meet Objective 3a through the selection and validation of a 5:1 gear ratio.

3.3.1 Selected Gear Geometry

For a fixed ring gear system, the reduction ratio is given by:

$$i = 1 + \frac{N_r}{N_s}, \tag{3.4}$$

Where N_s and N_r are the sun and ring gear teeth, respectively. Setting $i = 5$ gives:

$$N_r = 4N_s. \tag{3.5}$$

Considering the relation for a standard planetary set:

$$N_r = N_s + 2N_p, \tag{3.6}$$

Resulting:

$$4N_s = N_s + 2N_p \quad \Rightarrow \quad N_p = \frac{3}{2}N_s. \tag{3.7}$$

Choosing $N_s = 12$ yields $N_p = 18$ and $N_r = 48$. Other options would include:

$$\begin{array}{lll} \text{If } N_s = 10, & N_p = 15, & N_r = 40, \\ \text{If } N_s = 14, & N_p = 21, & N_r = 56. \end{array}$$

The modulus m, as defined by Moru and Borro (2020) in Equation 3.8:

$$m = \frac{d}{N} \tag{3.8}$$

For the ring gear with outer diameter $d_{\text{outer}} = m(N_r + 2)$ and a target of 90 mm:

$$50m = 110 \quad \Rightarrow \quad m = 1.8. \tag{3.9}$$

As $d_p = m \times Z$ (Kim et al., 2024) where m and Z are the tooth module and number respectively. The final pitch diameters are:

$$d_s = 1.8 \times 12 = 21.6\,\text{mm}, \quad d_p = 1.8 \times 18 = 32.4\,\text{mm}, \quad d_r = 1.8 \times 48 = 86.4\,\text{mm}. \tag{3.10}$$

3.3.2 Maximal Gear Stress

The maximal bending stress on the gear teeth can be approximated using the Lewis method, previously discussed in Section 2.2.4.1. This method will now be applied to the sun gear to meet Objective 3a. Assuming a 4.6195 N m motor torque and 8.55 mm radius the perpendicular force on the gear teeth F_t is:

$$F_t = \frac{4.6195\,\text{N m}}{8.55\,\text{mm}} = 540.29\,\text{N} \tag{3.11}$$

Load sharing across 3 teeth:

$$F_{\text{effective}} = \frac{F_t}{3} = \frac{540.29\,\text{N}}{3} \approx 180.1\,\text{N}. \tag{3.12}$$

Substituting numerical values in the Lewis formula from Equation 2.3,

$$\sigma = \frac{180.1\,\text{N}}{18\,\text{mm} \times 1.8\,\text{mm} \times 0.3}. \tag{3.13}$$

First compute the denominator:

$$b \cdot m \cdot Y = 18 \times 1.8 \times 0.3 \approx 9.72\,\text{mm}^2. \tag{3.14}$$

Hence,

$$\sigma \approx \frac{180.1\,\text{N}}{9.72\,\text{mm}^2} \approx 18.53\,\text{MPa} \tag{3.15}$$

The Ultimate Tensile Strength (UTS) (σ_{UTS}) for PA 12 is 53 MPa, leading to Equation 3.16

$$\text{Safety Factor} = \frac{\sigma_{UTS}}{\sigma} = \frac{53\,\text{MPa}}{18.53\,\text{MPa}} = 2.86 \tag{3.16}$$

Repeating this for the planet gear, assuming the force distribution on two teeth, one engaging with the sun gear and the other with the ring gear yields a Safety Factor of 1.29 calculated in Section A.2.

3.4 Lattice Properties Code Methodology

Larger versions of the plots outlined in this section containing 6 subplots are available in Section A.1. This section meets Objective 2 and only 3 subplots will be shown for clarity. The first step of the methodology is to create a TPMS, previously described in Equation 2.7, this is done through LatticeWorks. The option struct is set to have a $9 \times 3 \times 3$ cells which are the gyroid repetitions in the x, y and z directions. The creation of an automated FEA process for different level-sets between -1.3 and 1.3 Equation 2.4 is repeated between 64 iterations.

In Figure 3.1 the *isosurface* Matlab function is used in Section B.1 to generate the iso-surface in blue from the gyroid scalar field, using an adapted advancing front algorithm (Buchau and Rucker, 2017). Then *isocaps* generates cross sections at the boundary limits, this is necessary to create manifold geometry.

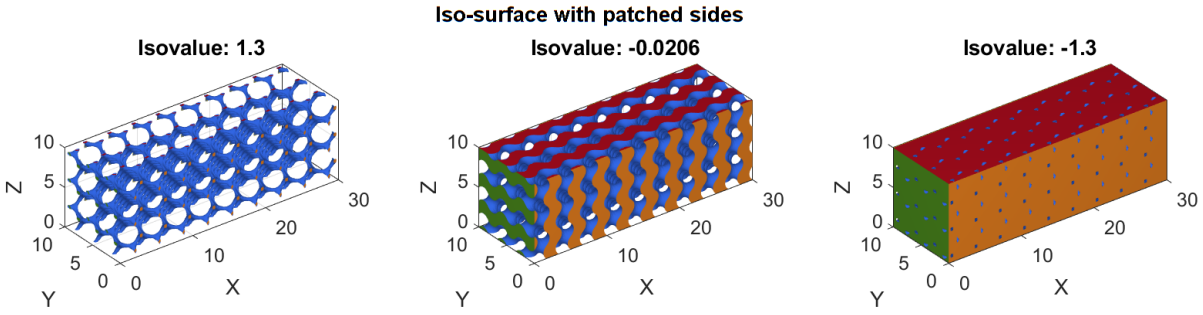


Fig. 3.1 Iso-surfaces with patched sides

Figure 3.2 shows the generated iso-surfaces at an orthographic side view.

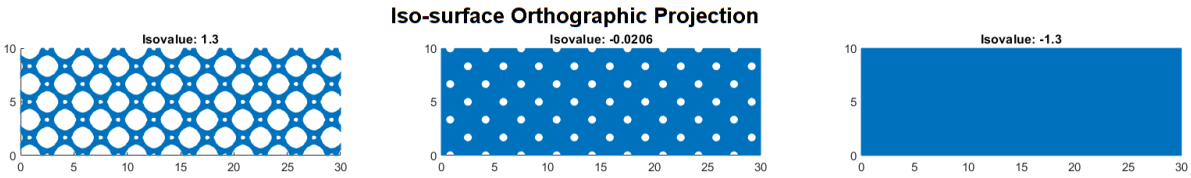


Fig. 3.2 Orthographic iso-surface view

The model is then meshed using TetGen (Section 2.6.1.1). Matlab array logic selects the node sets. The second half of the x length is selected from 15 to 30mm to not include the clamped region. As concluded by Evin et al. (2016), the deformation in the active region - away from clamping is most representative of the true material properties.

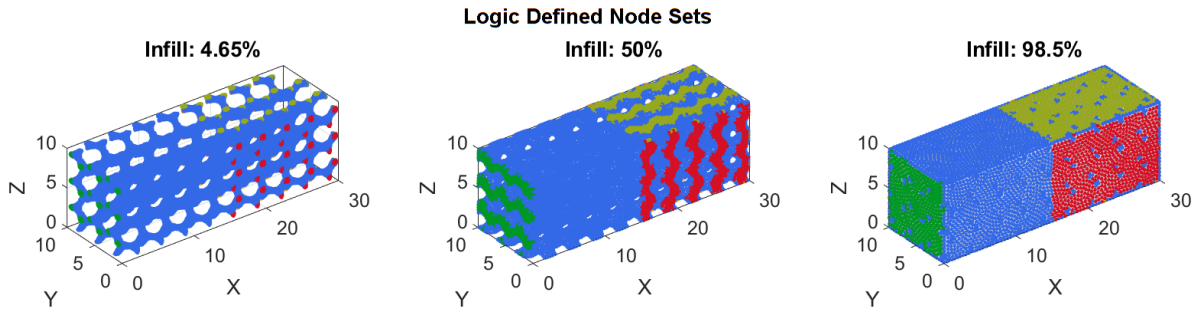


Fig. 3.3 Logic defined node sets

These node sets are then defined as shown in Figure 3.4. One side, coloured in black, is fixed. Whilst the red is displaced in the positive X direction. The displacements of the blue and cyan nodes are tracked.

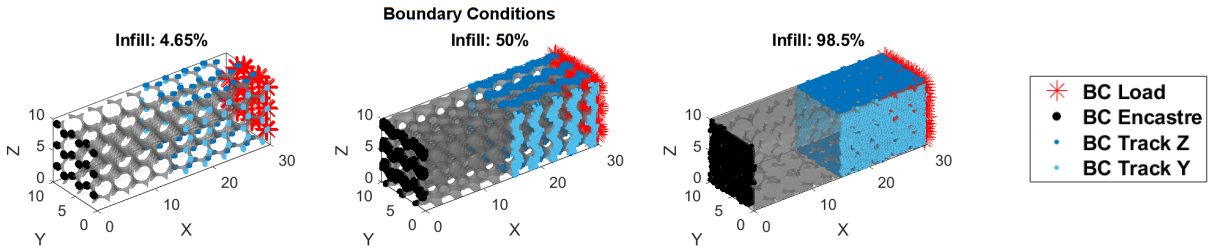


Fig. 3.4 Boundary conditions

Young's modulus is calculated using a linear estimation, and Poisson's ratio is also calculated at every timestep but is later averaged, this data is then discussed in Figures 4.1 and 4.2a.

Given for PA 12:$E = 1.666 \times 10^9$ Pa, $\nu = 0.38$, mm, $L = 30$ mm and testing with $\Delta L = 0.5$

$$\varepsilon_x = \frac{\Delta L}{L} = \frac{0.5}{30} = 0.01667 \tag{3.17}$$

Thus resulting in a distributed stress of:

$$\sigma = E \cdot \varepsilon_x = (1.666 \times 10^9) \times 0.01667 = 27.78 \times 10^6 \,\text{Pa} = 27.78 \,\text{MPa} \tag{3.18}$$

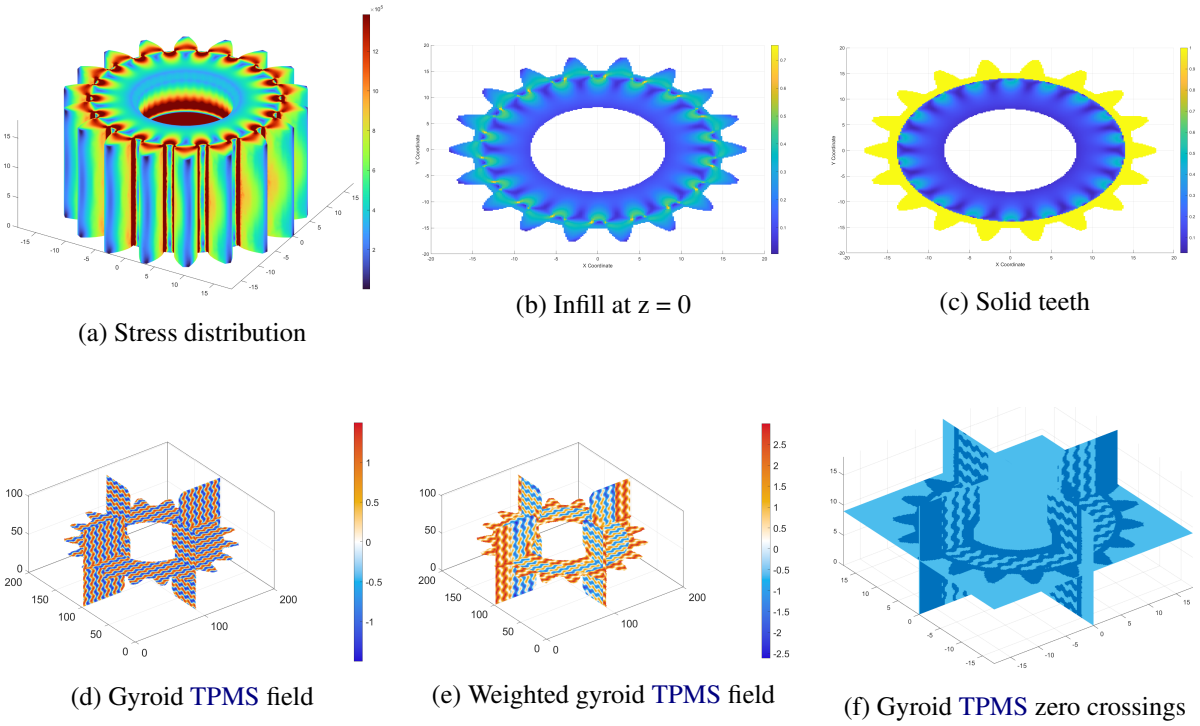
Thus justifying the use of a displacement of 0.5 mm in the FEM simulations.

3.5 Variable Infill Density Methodology

3.5.1 Stress Based Variable Infill Density

Figure 3.5 illustrates the stress driven weighted TPMS function. The main purpose of this script is to prove the feasibility of such an approach in generating 3D printable, manifold geometry with variable infill, thus providing a computationally inexpensive improved gear infill before applying TO.

The identical loading of every gear tooth assumption applied to Figure 3.5a, is taken to be based on the symmetry contained in the loading conditions and in the gear configuration. In gears, the configuration is typically of constant and symmetrical geometry that leads to nominally equal tooth stresses when loaded by a centrally applied torque. It is consistent with standard gear analysis and design practice for gear symmetry (Norton Robert, 2000; Shigley and Mischke, 2002). The resulting stress distribution at z = 0 then directly corresponds to an infill in Figure 3.5b.



(a) Stress distribution

(b) Infill at z = 0

(c) Solid teeth

(d) Gyroid TPMS field

(e) Weighted gyroid TPMS field

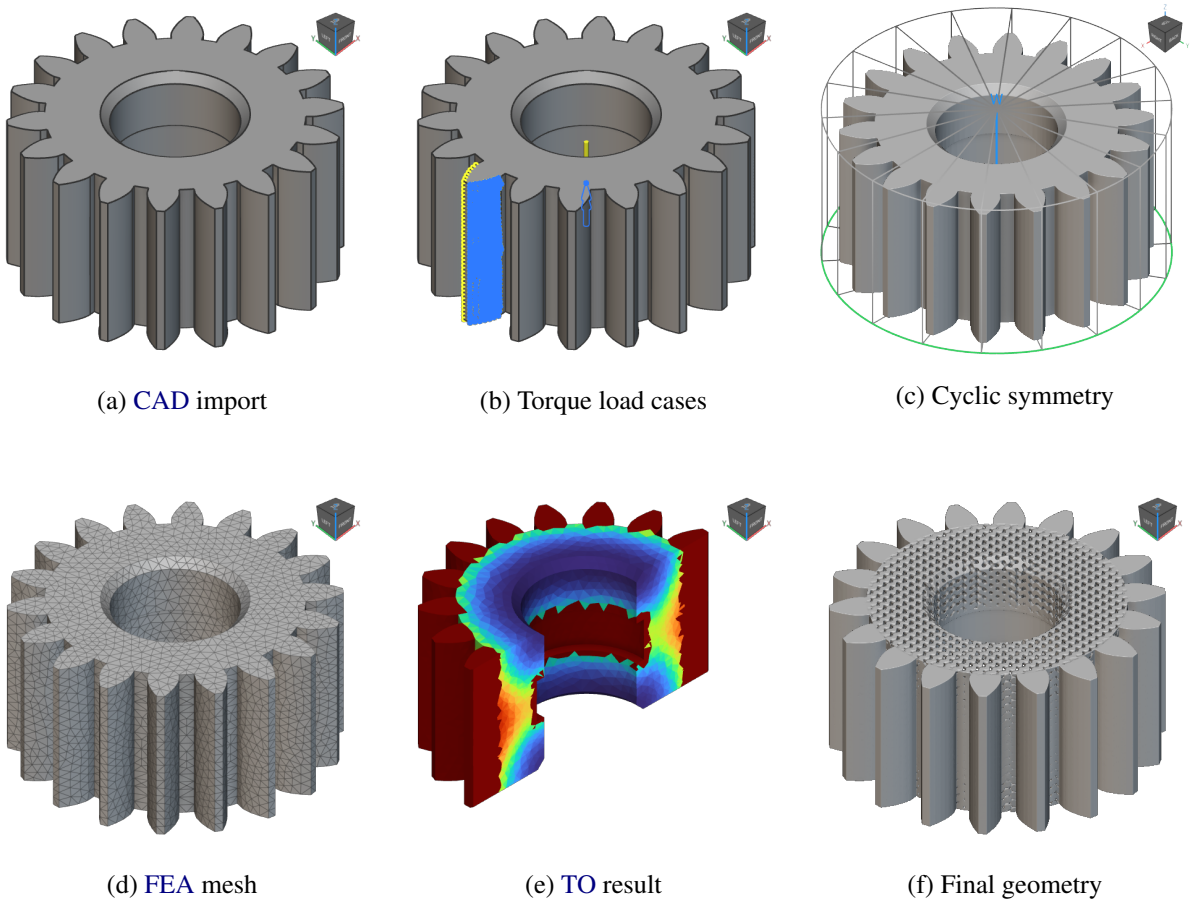
(f) Gyroid TPMS zero crossings

Fig. 3.5 Stress based infill distribution methodology

Figure 3.5d then shows the generated gyroid TPMS scalar field which would result in a 0 level-set mesh of 50% infill, which is later mapped to the stress distribution in Figure 3.5e. Given that the zero-crossings dictate what is solid when creating an iso-surface, these are visualised in Figure 3.5f. As expected, the area on the teeth has a solid infill.

3.5.2 TO Based Variable Infill Density

Figure 3.6 shows the nTopology TO approach. The generated Computer Aided Design (CAD) geometry is imported in Figure 3.6a, two loading steps apply a moment on each direction to account for back-driveability in Figure 3.6b - required for Objective 1b. A cyclic cellular map defines rotational symmetry in Figure 3.6c. The geometry is then meshed in Figure 3.6d and the TO result is shown in Figure 3.6e. Finally, in Figure 3.6f a TO scalar field drives the gyroid TPMS lattice density.



(a) CAD import (b) Torque load cases (c) Cyclic symmetry

(d) FEA mesh (e) TO result (f) Final geometry

Fig. 3.6 nTopology TO methodology

4. Results and Discussion

4.1 Matlab Method Summary Plots

Figure 4.1 shows the calculated Poisson's ratio at each infill. This is calculated in Section B.1 and is the average Poisson's ratio at each increment. The Abaqus solver creates an arbitrary amount of increments meaning several strains and elongations are found. These singular plots can be seen in Figure A.7, where the effective Poisson's ratio tends to slightly decrease with larger displacements.

At 20% infill, the connectivity of the gyroid TPMS structure is compromised meaning cells at the edges no longer fully connect. These floating elements are removed when meshing resulting in a decrease in the measured Poisson's ratio as disconnected elements are not load bearing (Maskery et al., 2018). This is especially evident when looking at the 4.65% infill case in Figure 4.9 and Figure 3.2.

The error is estimated using the Median Absolute Deviation (MAD) between the xy and xz median and mean which assumes a normal distribution (Wu et al., 2002). This approach is valid as the median is relatively unaffected by outliers. Re-running several simulations would give a higher confidence, reducing variance. However, since this takes over 15 hours of computing each time, this was not possible. The shaded region is within 1.96 standard deviations corresponding to a 95% confidence (Turner et al., 2023).

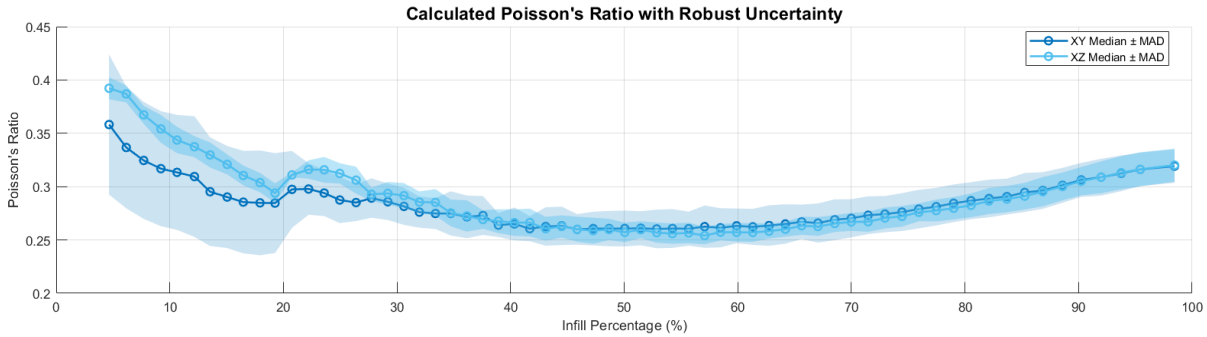
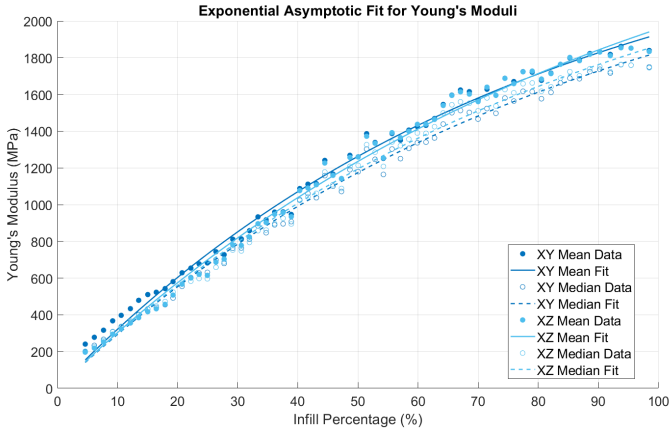


Fig. 4.1 Calculated Poisson's ratio against infill percentage

Due to the several increments the stress-strain relationship can be plotted, as shown for singular infill percentages in Figure A.7 and later summarised in Figure 4.2a. The data's trend is an asymptotic exponential which was fit using a custom non-linear regression model in Matlab. The increase in Young's modulus over 70% infill seems to have very diminishing returns. Devshette et al. (2023) comes to a similar conclusion where the difference between 60% and solid infill was considered negligible.

Figure 4.2a shows the Young's modulus asymptotic exponential model fit E using $E(x) = A - Ae^{-Bx}$



(a) Calculated Young's modulus

Fit Type	A	B	R^2
XY Mean	2670.8	0.0128	0.999864
XY Median	2658.6	0.0117	0.999897
XZ Mean	2973.9	0.0107	0.999876
XZ Median	2740	0.0114	0.999854

(b) Exponential asymptotic regression fit results.

Fig. 4.2 Determination of Young's modulus change with infill percentage

Figure 4.3 shows the R^2 values of Young's moduli are in the range of 0.9942 and 0.9952, indicating a strong linear fit and justifying the linear approximation approach previously used to find the Young's moduli plotted in Figure 4.2a. Once again, the effect of numerical instability and gyroid TPMS discontinuity is evident at lower infills. In this case, the mean approximation provides a more stable solution than the median. The individual stress-strain plots, which linearly approximated Young's moduli, are found in Figure A.9.

Zhang et al. (2018) stated that "deformation characteristics are mainly dependent on the unit cell geometries and are almost independent of relative density". However, this conclusion was based on a broader study of various TPMS structures. In contrast, my investigation, focused on gyroid TPMS infills, suggests a significant dependence of material properties on relative density, as demonstrated in Figure 4.1.

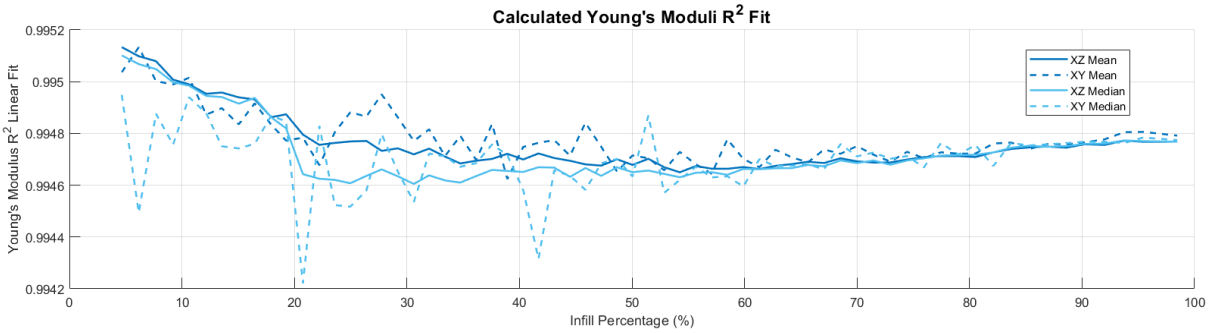
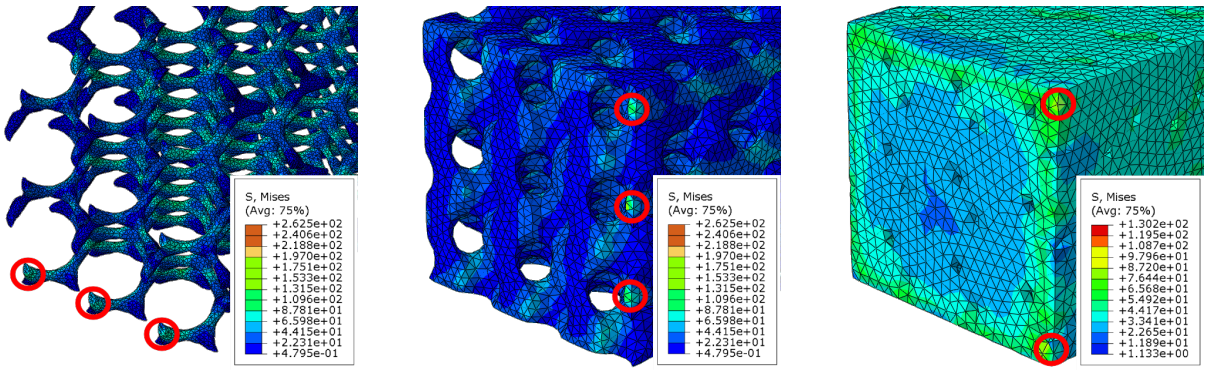


Fig. 4.3 Calculated Young's modulus linear fit R^2 against infill percentage

In Figure 4.4 the stress concentrations are circled in red. The maximum Young's modulus with a 100% infill is expected to be 1.666 GPa, as this was the material property definition used. The Young's modulus E is calculated at $x = 100$ using Equation 4.1 and coefficients from Figure 4.2b.

$$E(x) = A - Ae^{-Bx} \tag{4.1}$$

The $E_{XY\ Mean}, E_{XY\ Median}, E_{XZ\ Mean}, E_{XZ\ Median}$ equate to a Young's modulus at 100% infill of 1.928 GPa, 1.833 GPa, 1.954 GPa and 1.864 GPa respectively. The deviation of the actual value to the expected resulted in a percentage error of 15.73%, 10.02%, 17.29% and 11.88% respectively.



(a) Stress concentration at 4.65% (b) Stress concentration at 50% (c) Stress concentration at 98.5%

Fig. 4.4 Stress concentrations in Abaqus, concentrations circled in red

As the stress is calculated using the mean stress of the encastre fixed node list the stress concentrations result in a higher average stress cascading into an overestimate of the Young's modulus. The stress concentrations are caused by geometry and coarse meshing due to the 250,000 node limitation of the Abaqus academic license and distorted elements. In Figure 4.5, the distorted elements as a percentage of total elements is shown and appears to be consistent with a small 0.2% decrease under 10% infill indicating consistent meshing. A distorted element is defined by Abaqus as having an angle between isoparametric lines less than 45 degress of greater than 135 degrees. Distorted elements create mesh-distortion sensitivity which is the loss of polynomial completeness of shape functions under distorted element geometries (Rajendran, 2010).

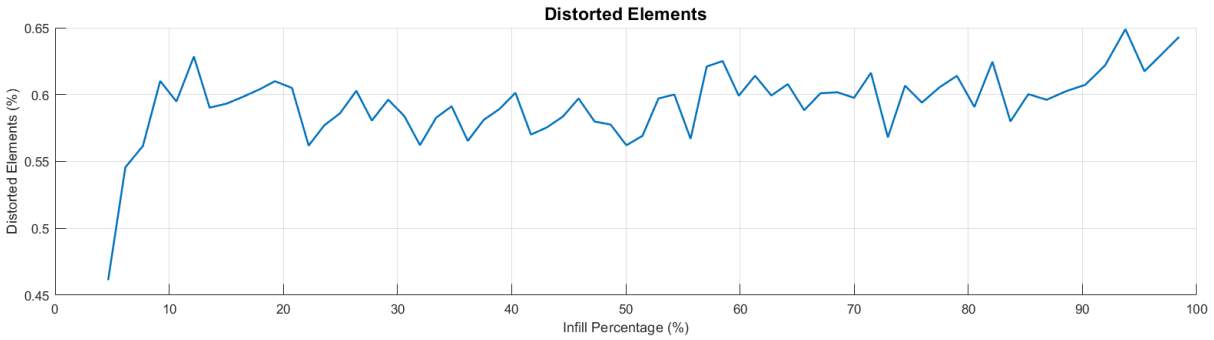


Fig. 4.5 Distorted elements against infill percentage

Figure 4.6 is relevant to calculate the contribution of simulations to the cost for Objective 3b. The area under the graph with an additional assumed 18W for peripherals and other components results in 2000.12 Wh summing to 35.8 pence at a rate of 17.9 pence per kWh (UK Government, 2024).

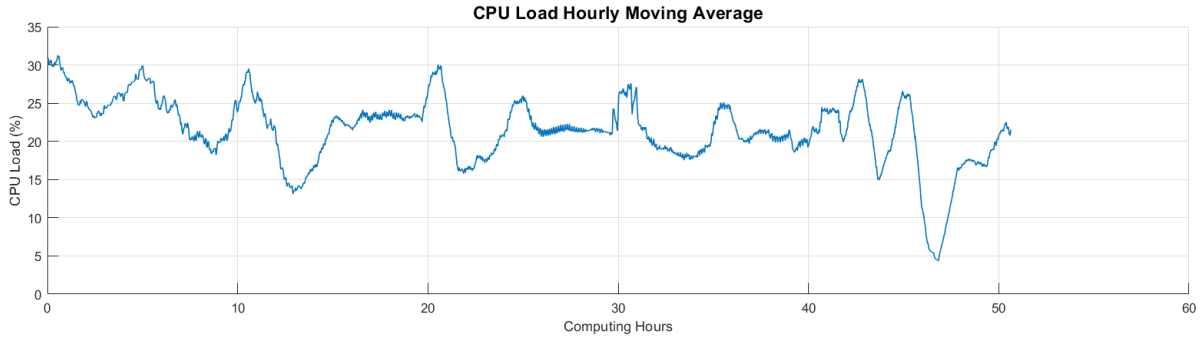


Fig. 4.6 CPU load hourly moving average

As the infill percentage increases the exposed surface at the boundaries where the nodes are tracked increases. Concurrently, since the mesh volume increases but the number of elements generated falls between 200 - 250 thousand the mesh is coarser. The step pattern in Figure 4.7 is due to the mesh being incrementally changed to be within this range. Since Tetgen cannot mesh to a target element count iterative incrementative meshing is used for all meshes to be in a narrow 50 thousand element range.

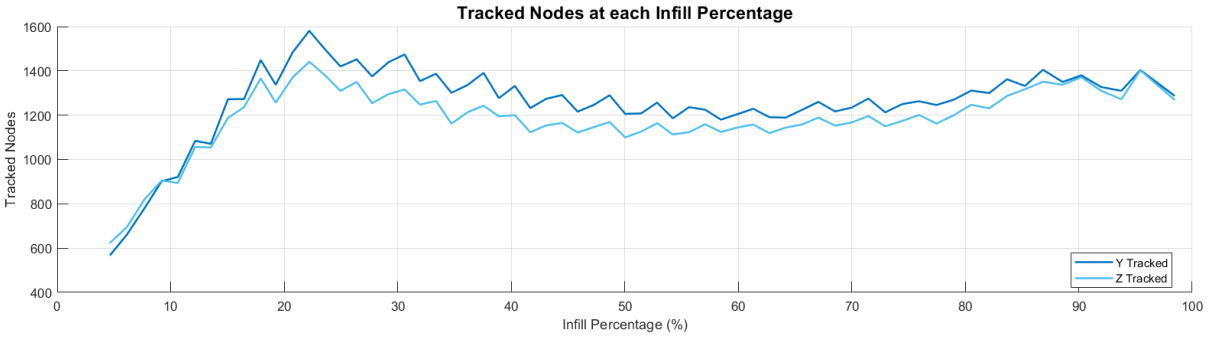


Fig. 4.7 Tracked nodes at each infill percentage

In Figure 4.8, the Von Mises stress at each infill percentage is shown. In the main code (Section B.1), displacement is set to 0.5mm and was purposely chosen to not exceed the previously identified UTS of 53 MPa for PA 12. Since a constant displacement is used as opposed to a constant force, higher infills are subjected to increased force due to a larger cross-sectional area for an identical displacement due to the additional tendency to resist deformation and evident by the increased Young's moduli in Figure 4.2a.

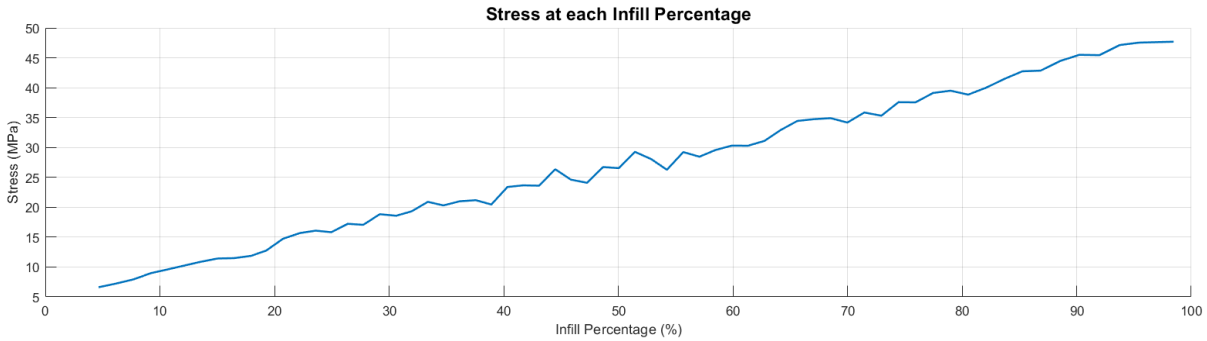
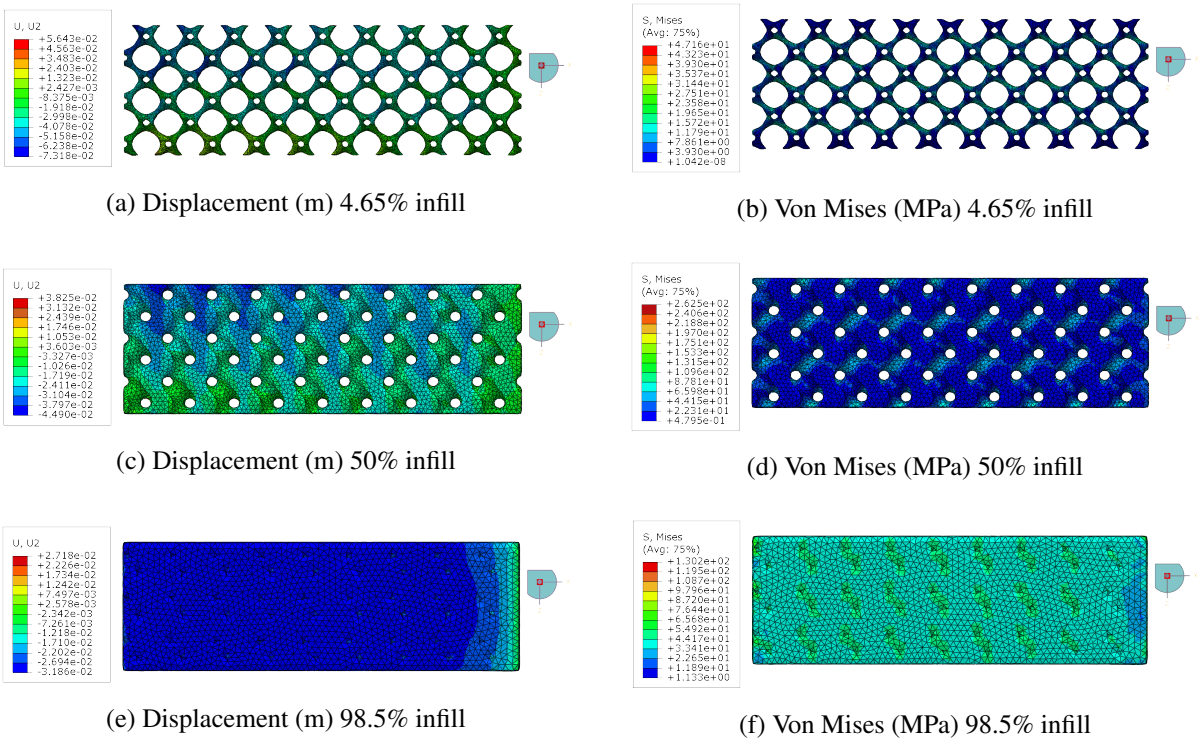


Fig. 4.8 Von Mises Stress at each infill percentage

Figure 4.9 illustrates the displacement and von Mises stress distributions for TPMS-based lattice structures at three different infill percentages: 4.65%, 50%, and 98.5%. The left column (Figure 4.9a, Figure 4.9c, Figure 4.9e) presents the displacement fields, while the right column (Figure 4.9b, Figure 4.9d, Figure 4.9f) shows the corresponding von Mises stress distributions.



(a) Displacement (m) 4.65% infill

(b) Von Mises (MPa) 4.65% infill

(c) Displacement (m) 50% infill

(d) Von Mises (MPa) 50% infill

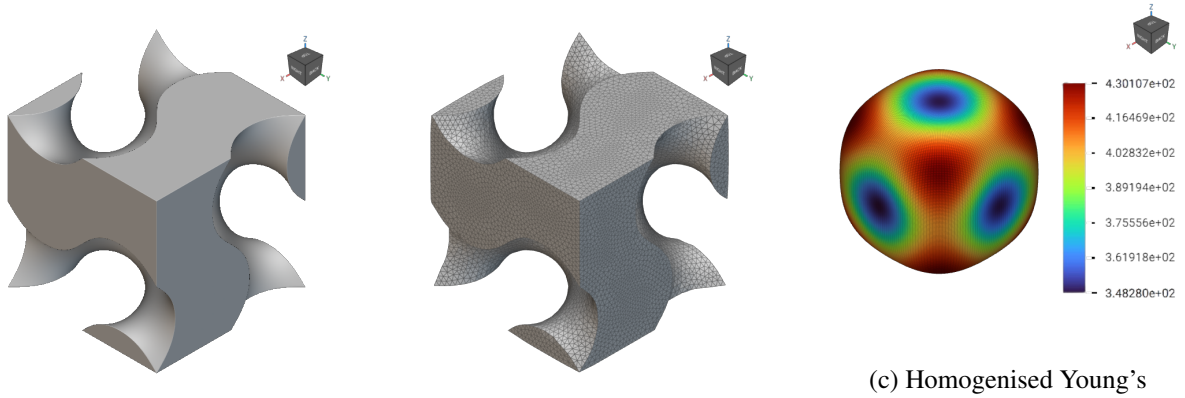
(e) Displacement (m) 98.5% infill

(f) Von Mises (MPa) 98.5% infill

Fig. 4.9 Lattice displacement and von Mises

4.2 TPMS Infill Homogenisation with nTopology

Figure 4.10 shows the homogenisation process, a single cell TPMS gyroid of size $3 \times 3 \times 3$ *mm* was generated then meshed using tetrahedral elements. Similarly to the previous Matlab method, the infill percentage was taken using the final mesh's volume in proportion to the 27 mm^2 boundary box size.



(a) Gyroid TPMS

(b) FEA mesh

(c) Homogenised Young's modulus at 50% infill (MPa)

Fig. 4.10 nTopology Homogenisation

Figure 4.11 depicts the values of the full stiffness matrix C in Voigt notation as previously discussed for Equation 3.2. As density increases (from 6.2% to 97.8%), the diagonal entries rise significantly, reflecting increased stiffness and load-carrying ability. Off-diagonal entries also rise but remain smaller, reflecting orthotropic behavior through direction-dependent stiffness that remains coupled. The effect is that low-density lattices are more compliant with high deformation and low stiffness, with higher-density lattices approaching the stiffness of the solid material of 1.666 GPa.

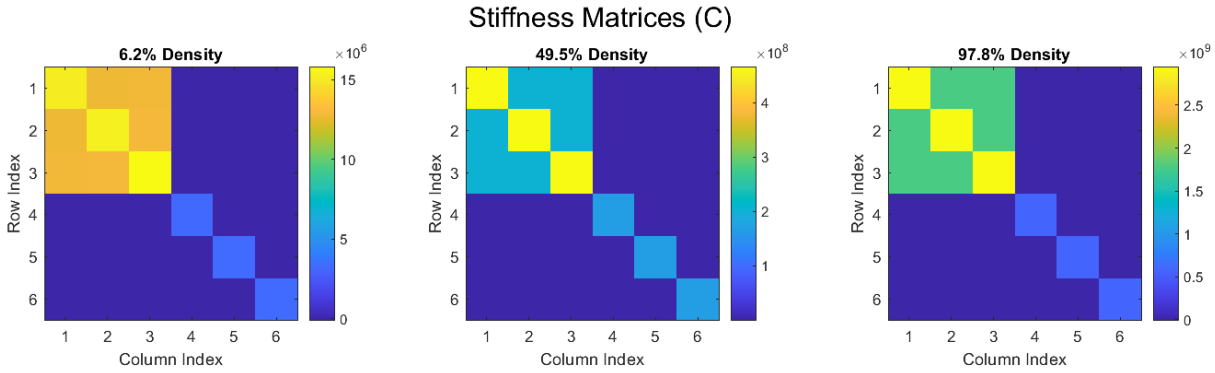


Fig. 4.11 Full stiffness matrix C visualisation heatmap in Pascals

Figure 4.12 shows an exponential increase of stiffness matrix C with infill density for the diagonal stiffness components. The diagonal stiffness components are visualised as they represent stiffness properties in the principal directions. On the other hand, the off-diagonal elements represent the coupling effects which weren't very significant. As any TPMS surface pattern including a TPMS gyroid, repeats in all three axis directions, the properties in the x, y and z directions are expected be the same or almost equal in all directions.

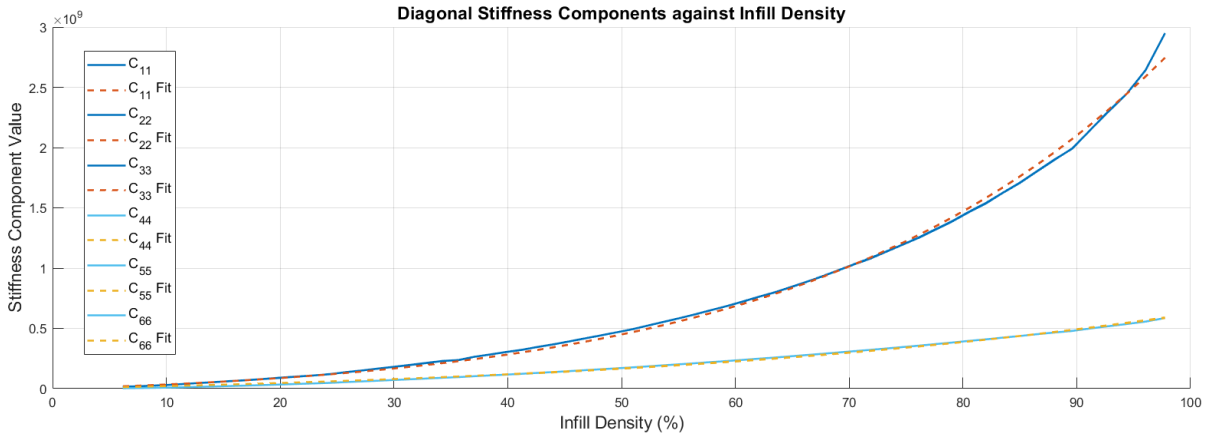


Fig. 4.12 Diagonal stiffness matrix components $C_{i=j}$

Figure 4.13 compares the two homogenisation results with the nTopology method using a single cell and the Matlab method using 81. To compare the nTopology and Matlab results the full 6×6 stiffness matrix C, previously visualised in Figure 4.11, was approximated assuming 0 non-diagonal entries and using Equation 3.3. While this results in a loss of anisotropic property definitions it is necessary for comparison and is justified by the relatively much smaller off-diagonal elements.

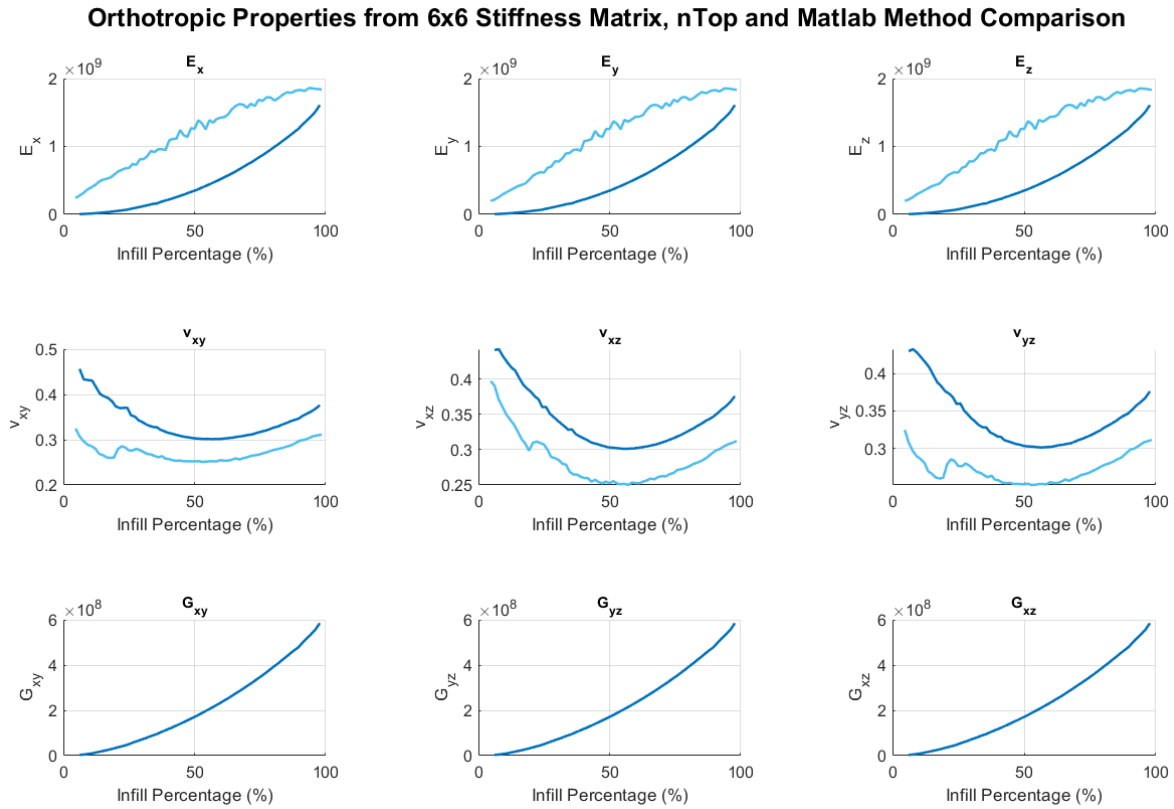


Fig. 4.13 nTop single cell orthotropic homogenisation approximation (blue) against Matlab $9 \times 3 \times 3$ cell size (cyan) in Pascals

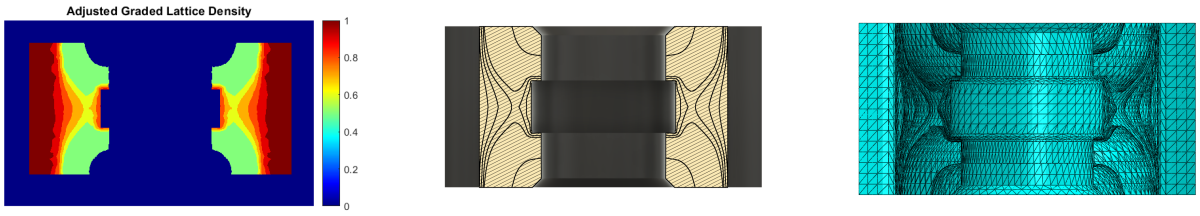
The discrepancy is not an error but rather the result of the simulation of a $9 \times 3 \times 3$ lattice and an isolated unit cell. The connected nature of TPMS lattices makes redistribution of imposed loads within the structure possible by the conversion of local stresses to bending moments in the lattice. The larger lattice structure thus has a lower Poisson's ratio than a free unit cell.

As stated by Wu et al. (2021a), "a general rule of thumb is that cells should be repeated 5 to 10 times before effective properties can be trusted", giving validity to the Matlab methodology as it used 81 cell repetitions.

The E_x, E_y and E_z equated to a Young's modulus at 97.76% (since this the highest simulated infill) of 1.61 GPa, 1.61 GPa and 1.6097 GPa respectively. The deviation of the actual value to the expected value of 1.666 GPa thus results in a percentage error of 3.36%, 3.36% and 3.38% respectively.

4.3 Optimised Gear Validation

Figure 4.14a shows a cross-section of the element density taken from Figure 3.6e which is then used to split a body with 6 discretised infill densities in Figure 4.14b. Once meshed in Figure 4.14c, 125,240 C3D10 tetrahedral mesh elements were generated, of which 9,620 were distorted due to the complex partitions in the mesh causing numerical singularities and eventual non-convergence. As this is caused by small elements between partition boundaries, a commercial Abaqus license with more mesh elements would likely converge. Validation by full-scale analysis in 2D only starts to appear in a handful of recent papers (Garner et al., 2019; Groen and Sigmund, 2018; Wu et al., 2021b)



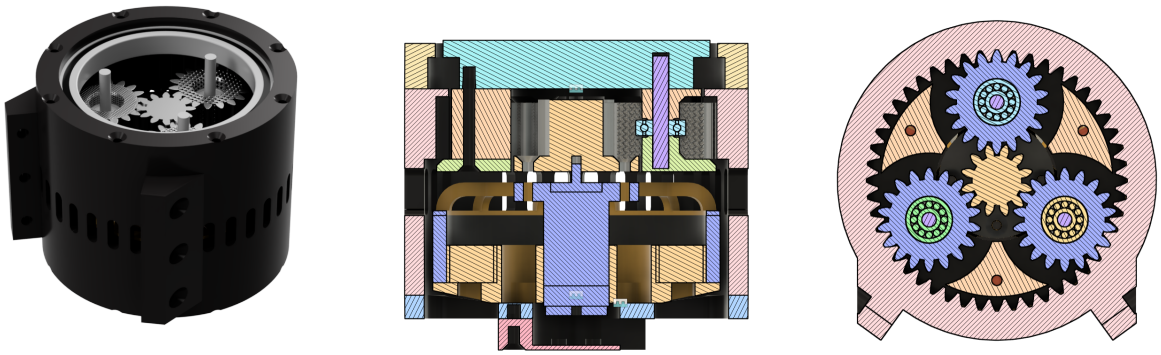
(a) Density gradient (b) Segmented gear model (c) Meshed gear for analysis

Fig. 4.14 Advanced meshing for gear validation

4.4 Final Design

4.4.1 Actuator Renders

Figure 4.15 shows the full actuator design. Figure 4.15a is an isometric top view with the top removed to reveal the carrier bearing and gear layout. Figure 4.15b depicts the internal layout and cross-section. Figure 4.15c highlights the gear assembly. The three planetary gears coloured in blue are fitted with bearings and dowels and the ring gear is incorporated into the case. The bearings were fit into the gears mid print allowing them to be permanently inserted without sliding. The complete design sketches are available in Appendix D.



(a) Isometric top view (b) X - Section view (c) Z - Section view

Fig. 4.15 Actuator design renders

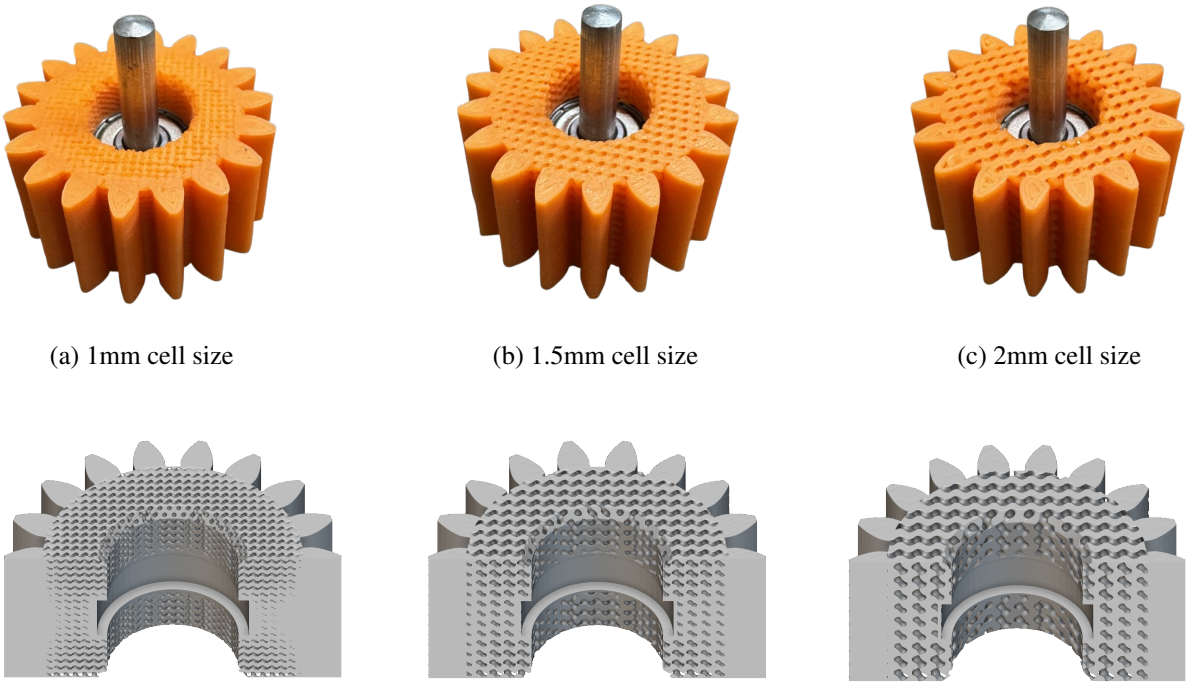
4.5 Final 3D Prints

The final actuator 3D print is shown in Figure 4.16, as the bearings were inserted during the print a socket to hold the bearing was used to insert the dowels and a soldering iron was used to add brass inserts.

(a) Final 3D print (b) Removed top carrier (c) Motor and sun gear

Fig. 4.16 Actuator 3D prints

The gear geometry was generated by using the Gras Solutions (2023) GF Gear Generator Fusion 360 plug-in, with the gear specifications chosen in Section 3.3, resulting in an initial volume of 11 427.3 mm^3. The final theoretical volumes for the 1mm, 1.5mm and 2mm cell size generated using nTop and the methodology discussed in Section 3.5.2 are respectively 9591.4 mm^3, 8892.82 mm^3 and 8547.81 mm^3, resulting in weight savings of 16.1%, 22.2% and 25.2% with minimal stiffness reduction. The comparison of the 3D print with the original geometry is shown in Figure 4.17.



(a) 1mm cell size (b) 1.5mm cell size (c) 2mm cell size

(d) 1mm cell size (e) 1.5mm cell size (f) 2mm cell size

Fig. 4.17 Comparison of 3D print to original model

5. Conclusion and Recommendations

5.1 Conclusion

In conclusion, 64 automated FEA simulations were performed in Abaqus using Matlab using the Windows command line. Amounting to over 50 hours of script and FEA simulation time encompassing 6 distinct versions of the proposed script in Section B.1 before it was finalised. In the process, more than 960 figures were generated per re-run and then collated through adaptively sampled subplots, as seen in Section A.1. The simulated lattices are $9 \times 3 \times 3$ cells and $30mm \times 10mm \times 10mm$ in the $x,\ y,\ z$ directions.

This was compared to a similar homogenisation approach with nTopology, a specialised software for additive manufacturing, using a single cell compared to the previous 81 $(9 \times 3 \times 3)$, which yielded a full 6×6 stiffness matrix C.

The Matlab approach using the Gibbon and LatticeWorks libraries interfacing with Abaqus suffered from several limitations: discontinuity between several boundary cells at infills below 20% caused a variation outside of expected values for Poisson's ratio that the nTopology approach did not have. Due to the automated meshing process used the elimination of distorted elements was not possible but was limited to below a moderate 0.65% of total elements. Distorted and coarse elements at the encastre boundary also contributed to a higher Young's modulus than expected through local stress concentrations.

5.1.1 Summary of Key Findings

The fitted equation for the $9 \times 3 \times 3$ cell Matlab simulation conducted in Section 4.1 for the XY Mean dataset are $E(x) = 2925.1 - 2860.1\,e^{-0.0107x}$. Similarly, the XY Median fit is represented by $E(x) = 2714.2 - 2700.7\,e^{-0.0112x}$. For the XZ datasets, the Mean fit is $E(x) = 2801.6 - 2843.7\,e^{-0.0120x}$, and the Median fit is $E(x) = 2681.9 - 2697\,e^{-0.0119x}$. Paired with a high R^2 of 0.999864, 0.999897, 0.999876 and 0.999854 respectively. At 100% infill, using the regression models fitted the $E_{XY\ Mean}$, $E_{XY\ Median}$, $E_{XZ\ Mean}$, $E_{XZ\ Median}$ equate to Young's moduli of 1.928 GPa, 1.833 GPa, 1.954 GPa and 1.864 GPa respectively. Causing a percentage error of 15.73%, 10.02%, 17.29% and 11.88% respectively.

In contrast, the nTopology homogenisation method conducted in Section 4.2 on a single $1 \times 1 \times 1$ cell yielded a 6×6 stiffness matrix C but was approximated to an orthotropic equivalent using the inverse of C (the flexibility matrix S) for comparison between the two methods. The E_x, E_y and E_z Young's moduli from the densest infill simulated were 1.61 GPa, 1.61 GPa and 1.6097 GPa respectively. The deviation of

the actual value to the expected value of 1.666 GPa thus results in a percentage error of 3.36%, 3.36% and 3.38% respectively.

The solid planet gears had a volume of 11 427.3 mm^3. The use of a topology optimised graded gyroid TPMS infill yielded three gear geometries of 9591.4 mm^3, 8892.82 mm^3 and 8547.81 mm^3, resulting in weight savings of 16.1%, 22.2% and 25.2% with minimal stiffness reduction. Since the cell size is independent of the topology optimisation result all of these solutions are valid, with the caveat that a higher cell size would exhibit more discretized local properties that will not be captured by the homogenised solution that was found.

5.1.2 Limitations

The safety factor of the sun gear through calculations was found to be 2.86 in Equation 3.16, due to the fact it is significantly smaller in mass than the three planet gears the sun gear was not optimised. It is also worth noting that the Lewis method provided the lower bound of the safety factor, while this allowed to focus more on TO, infill homogenisation and graded TPMS lattices, a simplified gear validation method is a limitation.

Some design choices of the proposed actuator based on the open-source OpenTorque design were not able to be fully validated in the length of this report. Due to the £100 limit imposed by Objective 3b several changes were made, outlined in Table 3.2 were not fully justified or validated with FEM but significantly reduced actuator price providing an opportunity for further work.

5.2 Recommendations for Future Work

A novel approach and method to homogenisation for characterisation of gyroid TPMS properties was proposed. The homogenised properties allow for the simulation of a variable infill density structure. While an FEA model was created in Section 4.3 the limitation of the 250,000 nodes resulted in bad and distorted elements due to the complex partitions required resulting in non-convergence of the model. While the creation of the optimised geometry has been thoroughly justified through the TO driven optimisation approach the result could not be fully validated. The explicit simulation using solid infill elements would require a mesh of over 5 million elements due to the complexity of the geometry which is unfeasible to generate, thus the homogenised results found provide a foundation for future research.

While the mechanical features of the proposed powered KO actuator have been well validated and discussed, further work needs to be done to design the brace that connects the actuator to the leg. Robust control algorithms could also be implemented to reduce backlash hysteresis as discussed in Section 2.2.5.2 using a rotary encoder and magnet. The implementation of a hard-stop is also required to not allow for knee over-extension causing injury in case of failure of control systems in scenarios such as gear slippage under load or temporary power outage to protect the user.

REFERENCES

N. K. Al-Hayali, S. M. Nacy, J. S. Chiad, and O. Hussein. Analysis and evaluation of a quasi-passive lower limb exoskeleton for gait rehabilitation. *Al-Khwarizmi Engineering Journal*, 2021. doi: 10.22153/kej.2021.12.007.

E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, Jan 2011. ISSN 1615-1488. doi: 10.1007/s00158-010-0594-7.

A. Barrera Sánchez, A. Blanco Ortega, E. Martínez Rayón, F. A. Gómez Becerra, A. Abúndez Pliego, R. Campos Amezcua, and C. H. Guzmán Valdivia. State of the art review of active and passive knee orthoses. *Machines*, 10(10), 2022. ISSN 2075-1702. doi: 10.3390/machines10100865.

M. P. Bendsøe and O. Sigmund. *Topology Optimization*. Springer, 2004. doi: 10.1007/978-3-662-05086-6.

P. Beyl, K. Knaepen, S. Duerinck, M. Van Damme, B. Vanderborght, R. Meeusen, and D. Lefeber. Safe and compliant guidance by a powered knee exoskeleton for robot-assisted rehabilitation of gait. *Advanced Robotics*, 25(5):513–535, 2011. doi: 10.1163/016918611X558225.

S. Bhardwaj, A. A. Khan, and M. Muzammil. Lower limb rehabilitation robotics: The current understanding and technology. *WORK: A Journal of Prevention, Assessment & Rehabilitation*, 69(3):775–793, 2021. doi: 10.3233/WOR-205012.

G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252, 2018. doi: 10.1109/IROS.2018.8593885.

M. Boolos, S. Corbin, A. Herrmann, and B. Regez. 3d printed orthotic leg brace with movement assist. *Annals of 3D Printed Medicine*, 7:100062, 2022. ISSN 2666-9641. doi: 10.1016/j.stlm.2022.100062.

B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001. doi: 10.1002/nme.116.

A. Buchau and W. Rucker. A meshfree isosurface computation method for boundary element methods. *International Journal of Computational Methods and Experimental Measurements*, 2017. doi: 10.2495/cmem-v5-n5-647-658.

A. R. Budarick, B. E. MacKeil, S. Fitzgerald, and C. D. Cowper-Smith. Design evaluation of a novel multicompartment unloader knee brace. *Journal of Biomechanical Engineering*, 142(1):014502, 11 2019. ISSN 0148-0731. doi: 10.1115/1.4044818.

C. Cai, W. S. Tey, J. Chen, W. Zhu, X. Liu, T. Liu, L. Zhao, and K. Zhou. Comparative study on 3d printing of polyamide 12 by selective laser sintering and multi jet fusion. *Journal of Materials Processing Technology*, 288:116882, 2021. ISSN 0924-0136. doi: 10.1016/j.jmatprotec.2020.116882.

X. Chen, H. Tang, J. Lin, and R. Zeng. Temporal trends in the disease burden of osteoarthritis from 1990 to 2019, and projections until 2030. *PLoS One*, 18(7):e0288561, July 2023. doi: 10.1371/journal.pone.0288561.

H. Chris-Amadin and O. Ibhadode. Lattgen: A tpms lattice generation tool. *Software Impacts*, 21:100665, 2024. ISSN 2665-9638. doi: 10.1016/j.simpa.2024.100665.

M. J. Claros, R. Soto, J. L. Gordillo, J. L. Pons, and J. L. Contreras-Vidal. Robotic assistance of human motion using active-backdrivability on a geared electromagnetic motor. *International Journal of Advanced Robotic Systems*, 13(2):40, 2016. doi: 10.5772/62331.

H. A. Dereshgi, E. Göse, D. Demir, and H. Ghannam. Restoring mobility and independence: Evaluating the impact of knee exoskeletons in real-world scenarios. *Journal of Smart Systems Research*, 2023. doi: 10.58769/joinssr.1308638.

A. Devshette, P. Mehta, S. Zambare, G. Patil, and V. Chaudhari. Modeling and fabrication of 5 – speed gearbox using 3d printing with optimum weight. *International Journal of Engineering Applied Sciences and Technology*, 8:213–219, 05 2023. doi: 10.33564/IJEAST.2023.v08i01.033.

S. Dinas and H. J. Martínez. *Delaunay Triangulation*, pages 1–6. Springer International Publishing, Cham, 2020. ISBN 978-3-319-08234-9. doi: 10.1007/978-3-319-08234-9_393-1.

D. Dudley and D. Townsend. *Dudley's Gear Handbook*. McGraw-Hill, 1991. ISBN 9780070179035.

Eaglepower, 2024. URL https://www.aliexpress.com/item/1005008122230642.html.

E. Evin, M. Tomáš, and M. Výrostek. Laser-beam welding impact on the deformation properties of stainless steels when used for automotive applications. *Acta Mechanica et Automatica*, 10(3):189–194, 2016. doi: 10.1515/ama-2016-0028.

R. J. Farris, H. A. Quintero, and M. Goldfarb. Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(6):652–659, 2011. doi: 10.1109/TNSRE.2011.2163083.

F. Ferrari and O. Sigmund. A new generation 99 line matlab code for compliance topology optimization and its extension to 3d. *Structural and Multidisciplinary Optimization*, 62, 08 2020. doi: 10.1007/s00158-020-02629-w.

E. Garner, H. M. Kolken, C. C. Wang, A. A. Zadpoor, and J. Wu. Compatibility in microstructural optimization for additive manufacturing. *Additive Manufacturing*, 26:65–75, 2019. ISSN 2214-8604. doi: 10.1016/j.addma.2018.12.007.

G. I. Giannopoulos. Linking md and fem to predict the mechanical behaviour of fullerene reinforced nylon-12. *Composites Part B: Engineering*, 161:455–463, 2019. ISSN 1359-8368. doi: 10.1016/j.compositesb.2018.12.110.

L. J. Gibson. Cellular solids. *Mrs Bulletin*, 28(4):270–274, 2003. doi: 10.1557/mrs2003.79.

Gras Solutions. Gf gear generator, 2023. URL https://apps.autodesk.com/FUSION/en/Detail/Index?id=1236778940008086660&appLang=en&os=Win64.

J. P. Groen and O. Sigmund. Homogenization-based topology optimization for high-resolution manufacturable microstructures. *International Journal for Numerical Methods in Engineering*, 113(8):1148–1163, 2018. doi: 10.1002/nme.5575.

G.-Y. Gu, L.-M. Zhu, and C.-Y. Su. Modeling and compensation of asymmetric hysteresis nonlinearity for piezoceramic actuators with a modified prandtl–ishlinskii model. *IEEE Transactions on Industrial Electronics*, 61(3):1583–1595, 2014. doi: 10.1109/TIE.2013.2257153.

W. He, D. Ofosu Amoateng, C. Yang, and D. Gong. Adaptive neural network control of a robotic manipulator with unknown backlash-like hysteresis. *IET Control Theory & Applications*, 11(4):567–575, 2017. doi: 10.1049/iet-cta.2016.1058.

A. Hendricks, S. Nevin, C. Wikoff, M. Dougherty, J. Orlita, and R. Noorani. The low-cost design and 3d printing of structural knee orthotics for athletic knee injury patients. *International Journal of Medical, Medicine and Health Sciences*, 11.0(10), Sept. 2018. doi: 10.5281/zenodo.1474763.

R. Homes, D. Clark, S. Moridzadeh, D. Tosovic, W. Van den Hoorn, K. Tucker, and M. Midwinter. Comparison of a wearable accelerometer/gyroscopic, portable gait analysis system (legsys+tm) to the laboratory standard of static motion capture camera analysis. *Sensors*, 23(1), 2023. ISSN 1424-8220. doi: 10.3390/s23010537.

S. F. Iftekar, A. Aabid, A. Amir, and M. Baig. Advancements and limitations in 3d printing materials and technologies: A critical review. *Polymers*, 15:2519, 05 2023. doi: 10.3390/polym15112519.

ISO. Plastics—Determination of Mechanical Properties in Static Tension—Part 1: General Principles. ISO 527:2012, 2012. ISO Standard.

F. Jabri, A. Ouballouch, L. Lasri, and R. El Alaiji. A comprehensive review of polymer materials and selective laser sintering technology for 3d printing. *Journal of Achievements in Materials and Manufacturing Engineering*, 118(1):5–17, May 2023. doi: 10.5604/01.3001.0053.7286.

M. Jung and G. Ludden. What do older adults and clinicians think about traditional mobility aids and exoskeleton technology? *ACM Transactions on Human-Robot Interaction*, 8:1–17, 06 2019. doi: 10.1145/3311789.

A. M. Kabe and B. H. Sako. Chapter 4 - model checks. In A. M. Kabe and B. H. Sako, editors, *Structural Dynamics Fundamentals and Advanced Applications*, pages 211–269. Academic Press, 2020. ISBN 978-0-12-821615-6. doi: 10.1016/B978-0-12-821615-6.00004-6.

V. Kandemir, O. Dogan, and U. Yaman. Topology optimization of 2.5d parts using the simp method with a variable thickness approach. *Procedia Manufacturing*, 17:29–36, 01 2018. doi: 10.1016/j.promfg.2018.10.009.

N. Karavas, A. Ajoudani, N. Tsagarakis, J. Saglia, A. Bicchi, and D. Caldwell. Tele-impedance based stiffness and motion augmentation for a knee exoskeleton device. In *2013 IEEE International Conference on Robotics and Automation*, pages 2194–2200, 2013. doi: 10.1109/ICRA.2013.6630872.

D. Kim, C. Lee, S. Hur, M. Choi, S. Shin, Y. Yang, and D. Choi. Development of 3d printed rapid omni-directional mecanum mobile manipulator. In *2021 18th International Conference on Ubiquitous Robots (UR)*, pages 124–129, 2021. doi: 10.1109/UR52253.2021.9494662.

S.-Y. Kim, Y.-D. Jun, S.-Y. Park, and B.-C. Go. Reverse engineering of geometric design parameters using shape information of non-standard spur gear. *Materialwissenschaft und Werkstofftechnik*, 55(5):641–649, 2024. doi: 10.1002/mawe.202400017.

E. Kirk-Wade, H. Wong, and S. Stiebahl. UK Disability Statistics: Prevalence and Life Experiences. Briefing Paper 9602, House of Commons Library, Oct 2024. URL https://researchbriefings.files.parliament.uk/documents/CBP-9602/CBP-9602.pdf.

P. Kshirsagar and N. Rathod. Artificial neural network. *International journal of computer applications*, 2:12–16, 2012. ISSN 0975-8887.

M. A. Laribi and S. Zeghloul. Chapter 4 - human lower limb operation tracking via motion capture systems. In M. Ceccarelli and G. Carbone, editors, *Design and Operation of Human Locomotion Systems*, pages 83–107. Academic Press, 2020. ISBN 978-0-12-815659-9. doi: 10.1016/B978-0-12-815659-9.00004-4.

G. Levine. Opentorque actuator: A powerful, compliant actuator for legged robotics, 2019. URL https://github.com/G-Levine/OpenTorque-Actuator.

S. C. Ligon, R. Liska, J. Stampfl, M. Gurr, and R. Mülhaupt. Polymers for 3d printing and customized additive manufacturing. *Chemical Reviews*, 117(15):10212–10290, Aug 2017. ISSN 0009-2665. doi: 10.1021/acs.chemrev.7b00074.

Mannion, S. Kaleb, D. Émélie, and Marc. A review of knee exoskeleton design aspects for improving user comfort. In *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, pages 1–7, 2024. doi: 10.1109/ICHMS59971.2024.10555841.

I. Maskery, A. Aremu, L. Parry, R. Wildman, C. Tuck, and I. Ashcroft. Effective design and simulation of surface-based lattice structures featuring volume fraction and cell type grading. *Materials & Design*, 155:220–232, 2018. ISSN 0264-1275. doi: 10.1016/j.matdes.2018.05.058.

J. R. Meda-Gutiérrez, L. A. Zúñiga-Avilés, A. H. Vilchis-González, and J. C. Ávila Vilchis. Knee exoskeletons design approaches to boost strength capability: A review. *Applied Sciences*, 11(21), 2021. ISSN 2076-3417. doi: 10.3390/app11219990.

G. Miller, D. Talmor, S.-h. Teng, N. Walkington, and H. Wang. Control volume meshes using sphere packing: Generation, refinement and coarsening. *Sandia National Laboratories*, 12 1999.

K. M. Moerman. Gibbon: The geometry and image-based bioengineering add-on. *Journal of Open Source Software*, 3(22):506, 2018. doi: 10.21105/joss.00506.

D. Moru and D. Borro. A machine vision algorithm for quality control inspection of gears. *The International Journal of Advanced Manufacturing Technology*, 106, 01 2020. doi: 10.1007/s00170-019-04426-2.

S. A. Murray, K. H. Ha, C. Hartigan, and M. Goldfarb. An assistive control approach for a lower-limb exoskeleton to facilitate recovery of walking following stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(3):441–449, 2015. doi: 10.1109/TNSRE.2014.2346193.

A. Musa. Openqdd v1: Open source 3d printed robotic actuator, 2023. URL https://github.com/aaedmusa/OpenQDD-V1.

C. W. Musser. Strain wave gearing, September 29 1959. URL https://patents.google.com/patent/US2906143A/en.

D. A. Neumann and E. R. Kelly. *Neumann's kinesiology of the musculoskeletal system*. Elsevier, 2025.

L. Norton Robert. Machine design an integrated approach, 2000.

A. Panesar, M. Abdi, D. Hickman, and I. Ashcroft. Strategies for functionally graded lattice structures derived using topology optimisation for additive manufacturing. *Additive Manufacturing*, 19, 11 2017. doi: 10.1016/j.addma.2017.11.008.

A. Pilipović, T. Brajlih, and I. Drstvenšek. Influence of processing parameters on tensile properties of sls polymer product. *Polymers*, 10(11), 2018. ISSN 2073-4360. doi: 10.3390/polym10111208.

L. Qi, D. Yang, B. Cao, Z. Li, and H. Liu. Design principle and numerical analysis for cycloidal drive considering clearance, deformation, and friction. *Alexandria Engineering Journal*, 91:403–418, 2024. ISSN 1110-0168. doi: 10.1016/j.aej.2024.01.077.

I.-C. Radu, A.-M. Vadureanu, D. Cozorici, E. Blanzeanu, and C. Zaharia. Advancing sustainability in modern polymer processing: Strategies for waste resource recovery and circular economy integration. *Polymers*, 17:522, 02 2025. doi: 10.3390/polym17040522.

S. Rajendran. A technique to develop mesh-distortion immune finite elements. *Computer Methods in Applied Mechanics and Engineering*, 199(17):1044–1063, 2010. ISSN 0045-7825. doi: 10.1016/j.cma.2009.11.017.

W. Roozing and G. Roozing. 3d-printable low-reduction cycloidal gearing for robotics. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1929–1935, 10 2022. doi: 10.1109/IROS47612.2022.9982006.

S. Rosso, R. Meneghello, L. Biasetto, L. Grigolato, G. Concheri, and G. Savio. In-depth comparison of polyamide 12 parts manufactured by multi jet fusion and selective laser sintering. *Additive Manufacturing*, 36:1–13, 12 2020. doi: 10.1016/j.addma.2020.101713.

E. Saerens, S. Crispel, P. Lopez Garcia, T. Verstraten, V. Ducastel, B. Vanderborght, and D. Lefeber. Scaling laws for robotic transmissions. *Mechanism and Machine Theory*, 140:601–621, 10 2019. doi: 10.1016/j.mechmachtheory.2019.06.027.

A. Salazar, A. Rico, J. Rodríguez, J. Segurado Escudero, R. Seltzer, and F. Martin de la Escalera Cutillas. Fatigue crack growth of sls polyamide 12: Effect of reinforcement and temperature. *Composites Part B: Engineering*, 59:285–292, 2014. ISSN 1359-8368. doi: 10.1016/j.compositesb.2013.12.017.

A. Sanchez, T. Rossos, A. Mihailidis, and B. Laschowski. Preliminary development of a robotic hip-knee exoskeleton with 3d-printed backdrivable actuators. *bioRxiv*, 2023. doi: 10.1101/2023.05.18.541377.

R. Sankineni and R. Yennam. Evaluation of energy absorption capabilities and mechanical properties in fdm printed pla tpms structures. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236: 095440622110395, 08 2021. doi: 10.1177/09544062211039530.

K. B. Shelburne, M. G. Pandy, and M. R. Torry. Comparison of shear forces and ligament loading in the healthy and ACL-deficient knee during gait. *J Biomech*, 37(3):313–319, Mar. 2004. doi: 10.1016/j.jbiomech.2003.07.001.

J. Shigley and C. Mischke. *Mechanical Engineering Design*. McGraw-Hill classic textbook reissue series. McGraw-Hill, 2002. ISBN 9780072832099.

H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2), Feb. 2015. ISSN 0098-3500. doi: 10.1145/2629697.

O. Sigmund. Sigmund, o.: A 99 line topology optimization code written in matlab. structural and multidisciplinary optimization 21, 120-127. *Structural and Multidisciplinary Optimization*, 21:120–127, 04 2001. doi: 10.1007/s001580050176.

F. Sillani, R. G. Kleijnen, M. Vetterli, M. Schmid, and K. Wegener. Selective laser sintering and multi jet fusion: Process-induced modification of the raw materials and analyses of parts performance. *Additive Manufacturing*, 27:32–41, 2019. ISSN 2214-8604. doi: 10.1016/j.addma.2019.02.004.

S. Singh, S. Ramakrishna, and F. Berto. 3d printing of polymer composites: A short review. *Material Design & Processing Communications*, 2(2):e97, 2020. doi: 10.1002/mdp2.97. e97 MDPC-2019-037.R1.

L. Siva Rama Krishna, N. Mahesh, and N.Sateesh. Topology optimization using solid isotropic material with penalization technique for additive manufacturing. *Materials Today: Proceedings*, 4(2, Part A):1414–1422, 2017. ISSN 2214-7853. doi: 10.1016/j.matpr.2017.01.163. 5th International Conference of Materials Processing and Characterization (ICMPC 2016).

Soham, R. Nevgi, V. Trivedi, H. Thasale, N. Siddhesh, B. Padate, and B. Mashilkar. Design and development of cycloidal gear box. In *International Conference on Recent Advances in Science, Engineering and Technology*, 06 2024.

D. Systèmes. Abaqus 2016 documentation: Element library overview, 2016. URL http://130.149.89.49:2080/v2016/books/usb/default.htm?startat=pt06ch27s01abo25.html.

T. Tang, H. Jia, J. Li, J. Wang, and X. Zeng. Modeling of transmission compliance and hysteresis considering degradation in a harmonic drive. *Applied Sciences*, 11(2), 2021. ISSN 2076-3417. doi: 10.3390/app11020665.

X. Tang, X. Wang, X. Ji, Y. Zhou, J. Yang, Y. Wei, and W. Zhang. A wearable lower limb exoskeleton: Reducing the energy cost of human movement. *Micromachines (Basel)*, 13(6), June 2022. doi: 10.3390/mi13060900.

S. Thumsorn, W. Prasong, T. Kurose, A. Ishigami, Y. Kobayashi, and H. Ito. Rheological behavior and dynamic mechanical properties for interpretation of layer adhesion in fdm 3d printing. *Polymers*, 14:2721, 07 2022. doi: 10.3390/polym14132721.

C. Torcătoru and A.-C. Tulică. A 3d-printed functional knee orthosis for daily activities. In L. Moldovan and A. Gligor, editors, *The 17th International Conference Interdisciplinarity in Engineering*, pages 129–141, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-54664-8.

U. Trivedi and A. Y. Joshi. Advances in active knee brace technology: A review of gait analysis, actuation, and control applications. *Heliyon*, 10(4), Feb 2024. ISSN 2405-8440. doi: 10.1016/j.heliyon.2024.e26060.

S. L. Turner, E. Korevaar, M. S. Cumpston, R. Kanukula, A. B. Forbes, and J. E. McKenzie. Effect estimates can be accurately calculated with data digitally extracted from interrupted time series graphs. *Research Synthesis Methods*, 14(4):622–638, 2023. doi: 10.1002/jrsm.1646.

UK Government, 2024. URL https://assets.publishing.service.gov.uk/media/6762955acdb5e64b69e30703/quarterly-energy-prices-december-2024.pdf.

M. Vafaee. Mahtab-vafaee/latticeworks: A matlab toolbox for functionally graded lattice generation, in different configurations., 2025. URL https://github.com/mahtab-vafaee/LatticeWorks/tree/main.

M. Vafaeefar, K. M. Moerman, and T. J. Vaughan. Latticeworks: An open-source matlab toolbox for nonuniform, gradient and multi-morphology lattice generation, and analysis. *Materials & Design*, 250:113564, 2025. ISSN 0264-1275. doi: 10.1016/j.matdes.2024.113564.

B. Van Hooreweder and J.-P. Kruth. High cycle fatigue properties of selective laser sintered parts in polyamide 12. *CIRP Annals*, 63(1):241–244, 2014. ISSN 0007-8506. doi: 10.1016/j.cirp.2014.03.060.

T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59(4):257–263, Sep 1988. ISSN 1432-0770. doi: 10.1007/BF00332914.

C. Wu, Y. Zhao, and Z. Wang. The median absolute deviations and their applications to shewhart (x)over-bar control charts. *Communication in Statistics- Simulation and Computation*, 3131:425–442, 01 2002. doi: 10.1081/SAC-120003850.

J. Wu, O. Sigmund, and J. P. Groen. Topology optimization of multi-scale structures: a review. *Structural and Multidisciplinary Optimization*, 63(3):1455–1480, Mar 2021a. ISSN 1615-1488. doi: 10.1007/s00158-021-02881-8.

J. Wu, W. Wang, and X. Gao. Design and optimization of conforming lattice structures. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):43–56, Jan. 2021b. ISSN 2160-9306. doi: 10.1109/tvcg.2019.2938946.

K. Xiao, F. Yuan, J. Wang, L. Li, Y. Han, C. Wang, and X. Chen. Spatial mapping tooth profile design and transmission characteristics analysis of harmonic drive based on scaling factor. *Meccanica*, 02 2025. doi: 10.1007/s11012-024-01933-9.

Z. Xu, Y. Wang, D. Wu, K. P. Ananth, and J. Bai. The process and performance comparison of polyamide 12 manufactured by multi jet fusion and selective laser sintering. *Journal of Manufacturing Processes*, 47:419–426, 2019. ISSN 1526-6125. doi: 10.1016/j.jmapro.2019.07.014.

T. Yarlagadda, Z. Zhang, L. Jiang, P. Bhargava, and A. Usmani. Solid isotropic material with thickness penalization – a 2.5d method for structural topology optimization. *Computers & Structures*, 270:106857, 07 2022. doi: 10.1016/j.compstruc.2022.106857.

T. Yoshida, G. Endo, A. Okubo, and H. Nabae. Experimental evaluation of a quasi-direct-drive actuator with a 3d-printed planetary gear reducer. In *2023 IEEE/SICE International Symposium on System Integration (SII)*, pages 1–6, 2023. doi: 10.1109/SII55687.2023.10039274.

A. Zakręcki, J. Cieślik, A. Bazan, and P. Turek. Innovative approaches to 3d printing of pa12 forearm orthoses: A comprehensive analysis of mechanical properties and production efficiency. *Materials*, 17(3), 2024. ISSN 1996-1944. doi: 10.3390/ma17030663.

L. Zhang, S. Feih, S. Daynes, S. Chang, M. Y. Wang, J. Wei, and W. F. Lu. Energy absorption characteristics of metallic triply periodic minimal surface sheet structures under compressive loading. *Additive Manufacturing*, 23:505–515, 2018. ISSN 2214-8604. doi: 10.1016/j.addma.2018.08.007.

Y. Zhang, K. Mao, S. Leigh, A. Shah, Z. Chao, and G. Ma. A parametric study of 3d printed polymer gears. *The International Journal of Advanced Manufacturing Technology*, 107(11):4481–4492, Apr 2020. ISSN 1433-3015. doi: 10.1007/s00170-020-05270-5.

A. Zhu, Y. Tanaka, F. Rafeedi, and D. Hong. Cycloidal quasi-direct drive actuator designs with learning-based torque estimation for legged robotics, 2024.

H. Zhu, C. Nesler, N. Divekar, V. Peddinti, and R. D. Gregg. Design principles for compact, backdrivable actuation in Partial-Assist powered knee orthoses. *IEEE ASME Trans Mechatron*, 26(6):3104–3115, Jan. 2021. doi: 10.1109/TMECH.2021.3053226.

O. Zienkiewicz, R. Taylor, and J. Zhu. Chapter 6 - shape functions, derivatives, and integration. In O. Zienkiewicz, R. Taylor, and J. Zhu, editors, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, pages 151–209. Butterworth-Heinemann, Oxford, seventh edition edition, 2013. ISBN 978-1-85617-633-0. doi: 10.1016/B978-1-85617-633-0.00006-X.

A. Extended Plots and Calculations

A.1 Expanded Methodology Plots

A total of 64 simulations were conducted

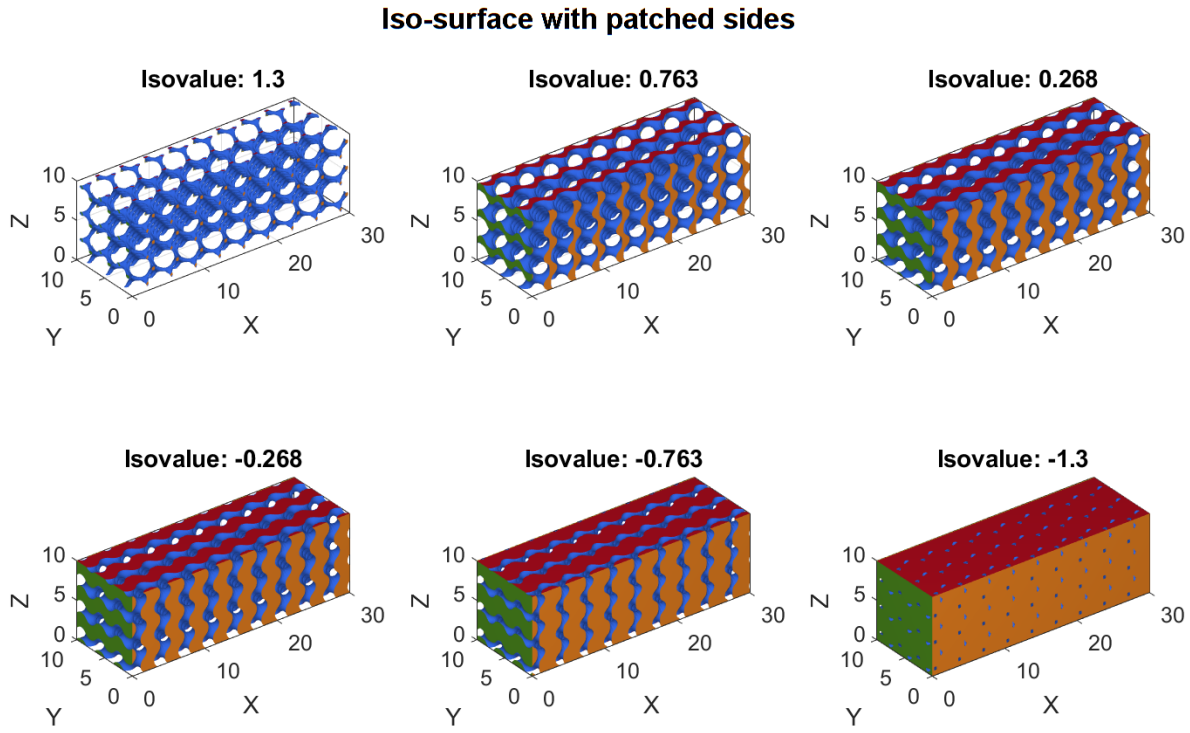


Fig. A.1 Iso-surfaces with patched sides

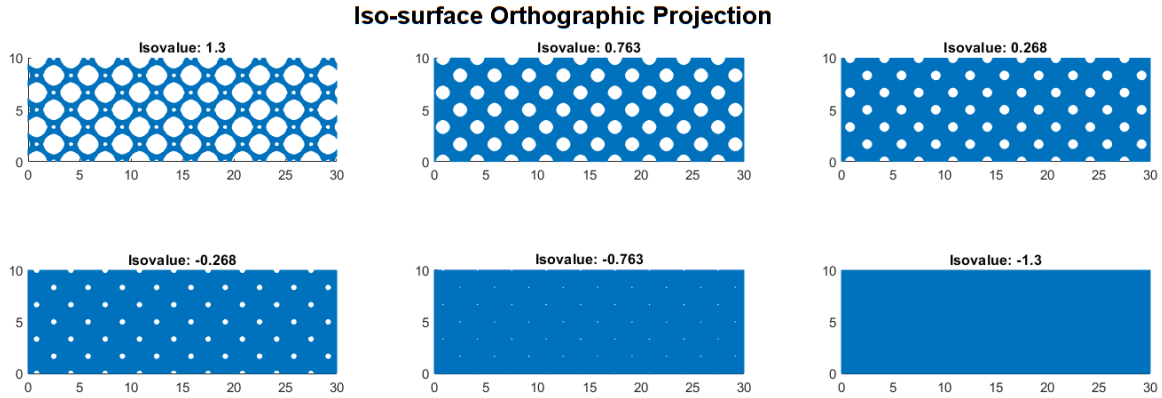


Fig. A.2 Orthographic iso-surface view

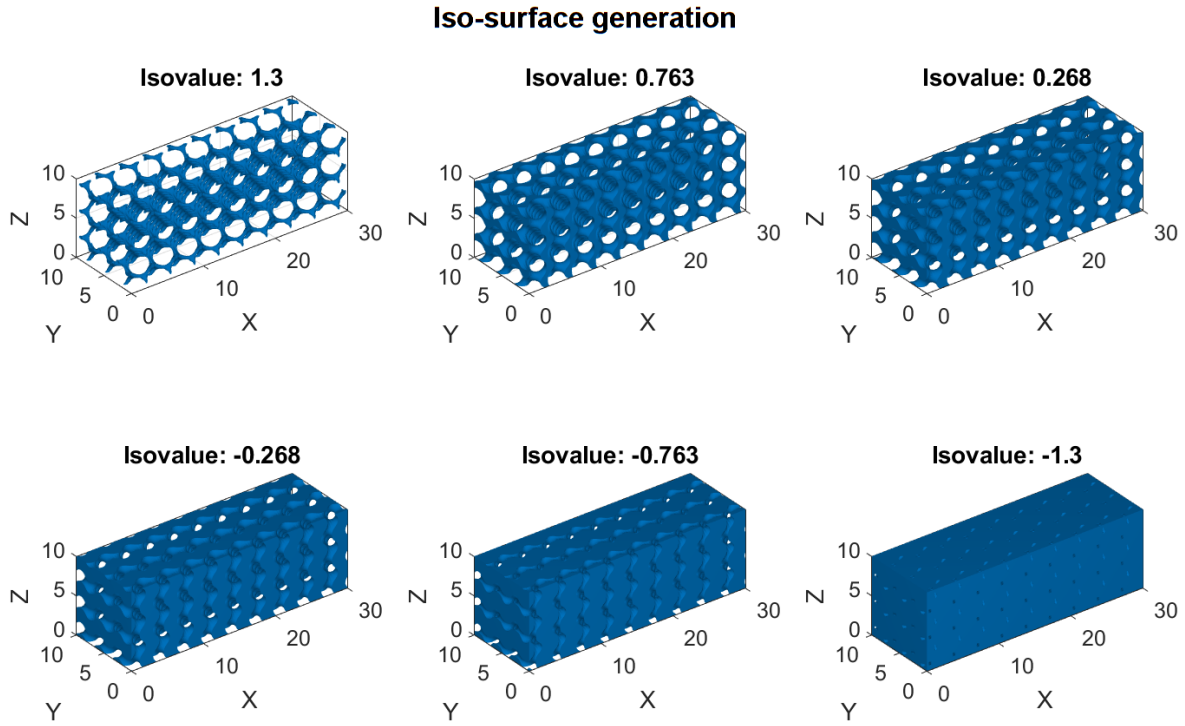


Fig. A.3 Generated iso-surface

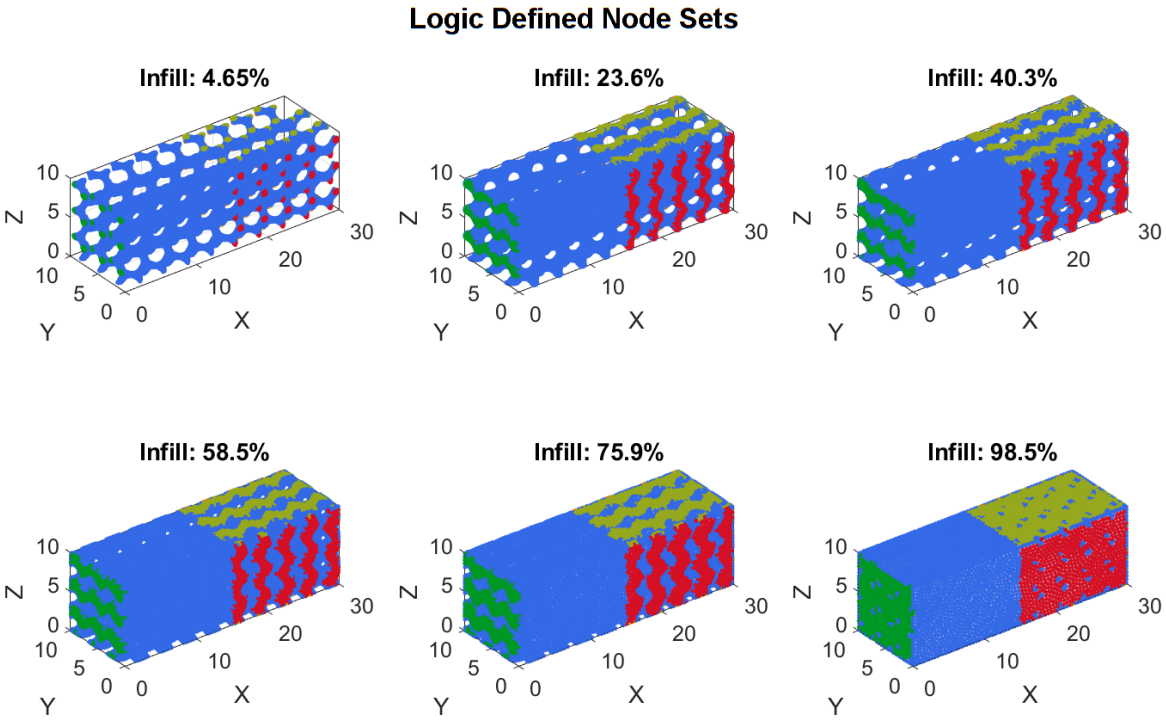


Fig. A.4 Logic defined node sets

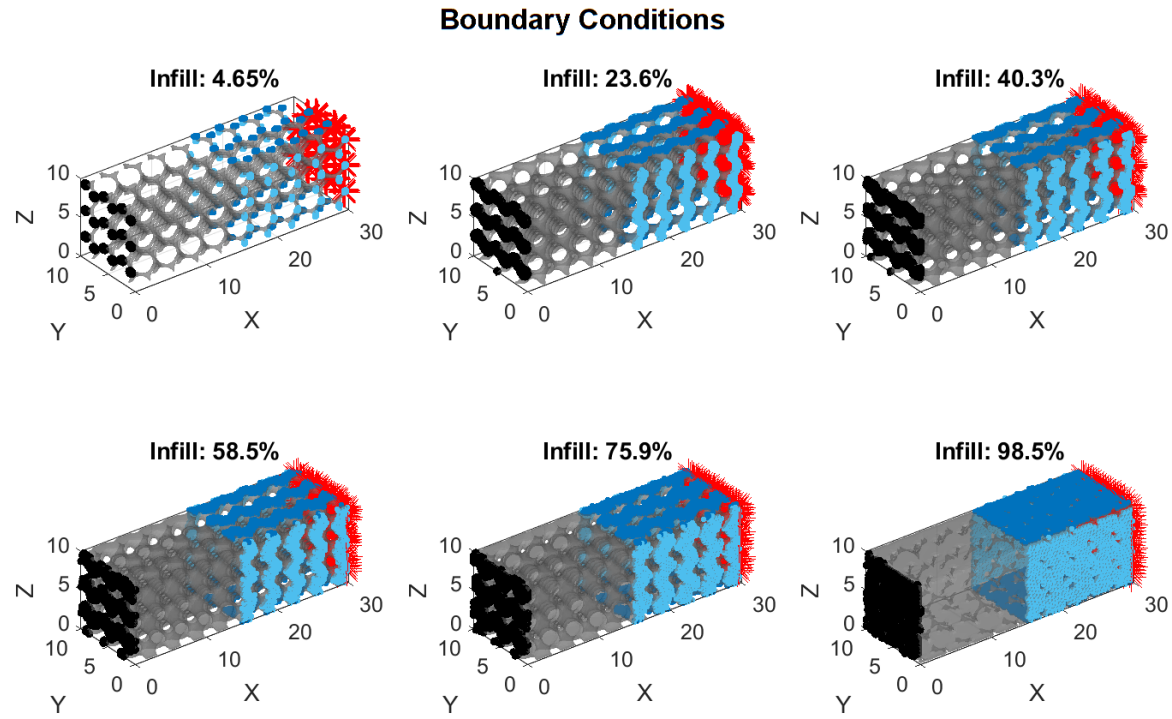


Fig. A.5 Boundary conditions

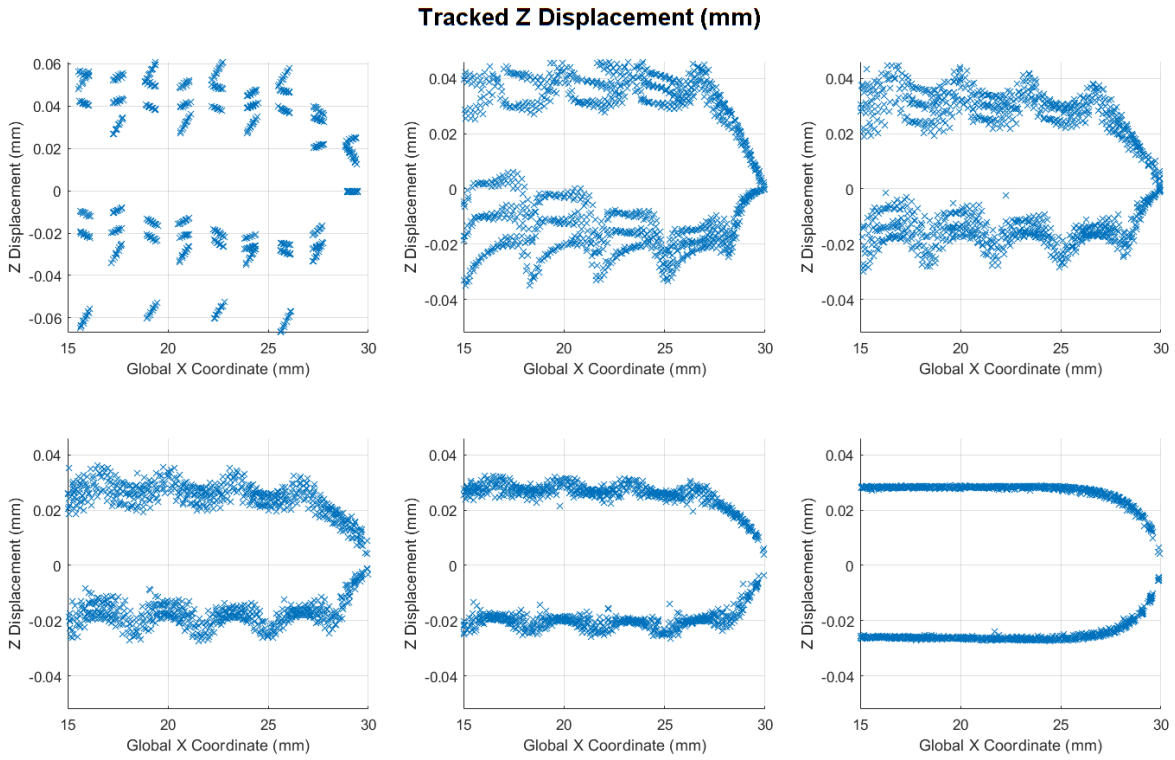


Fig. A.6 Z displacement at different infills

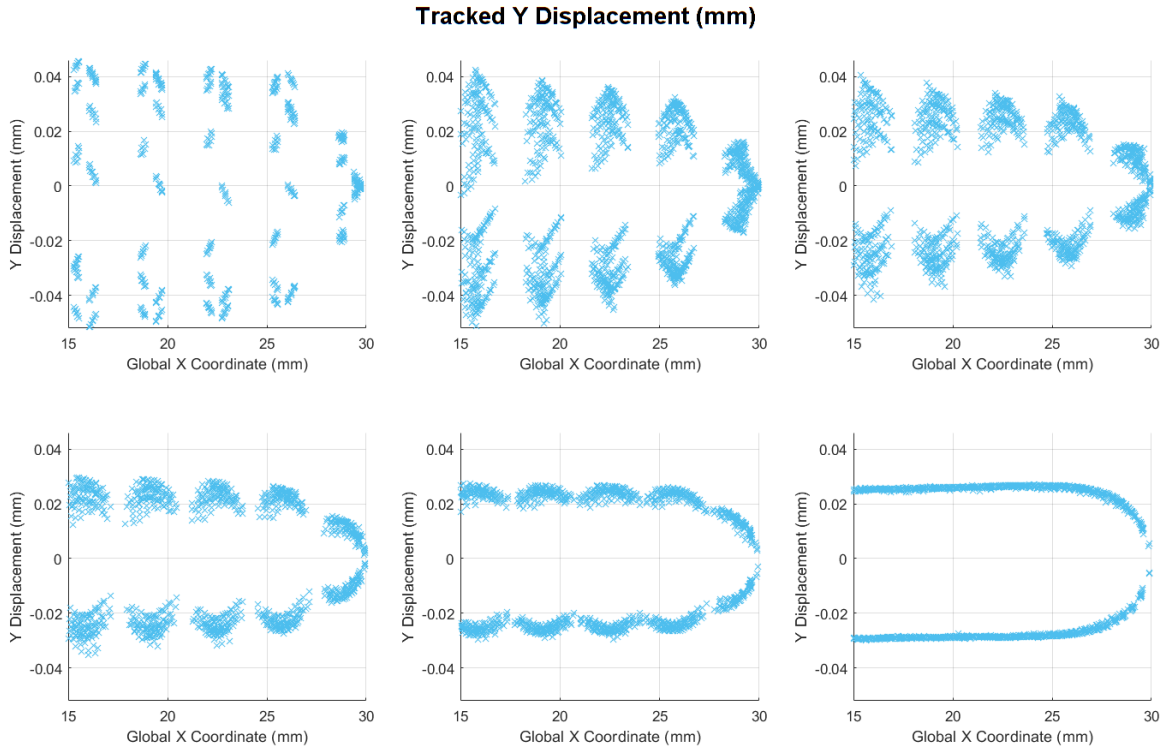


Fig. A.7 Y displacement at different infills

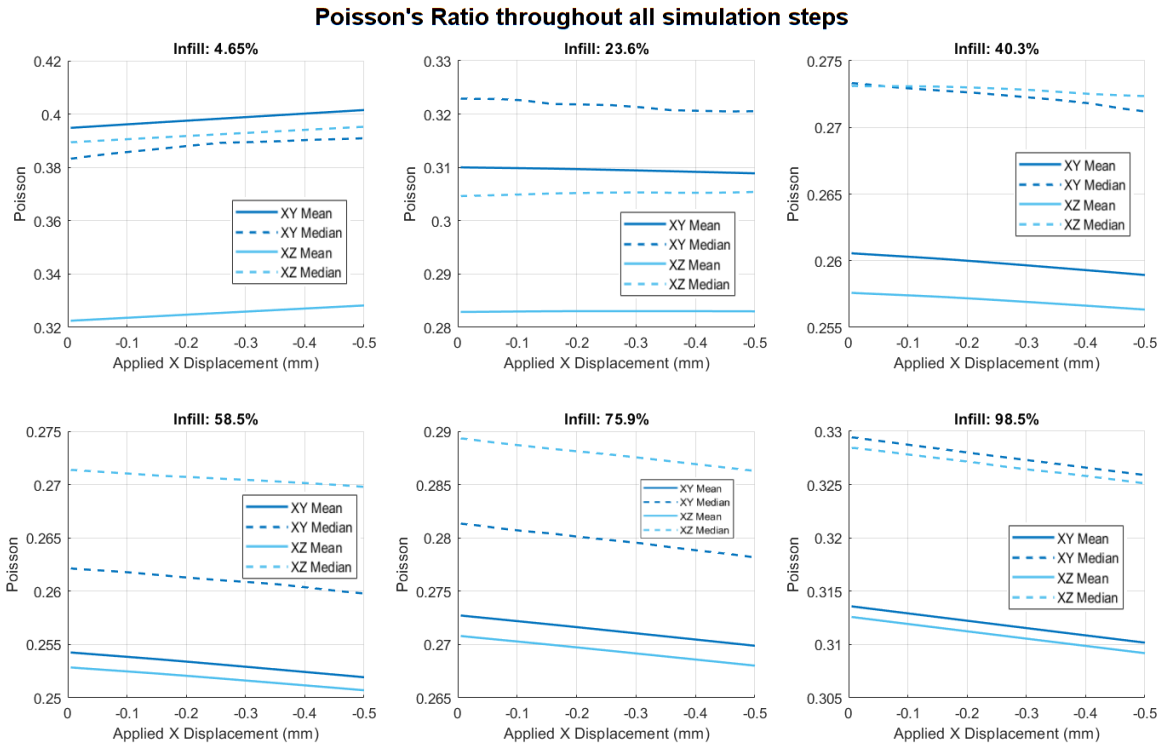


Fig. A.8 Poisson's ratio at different infills and timesteps

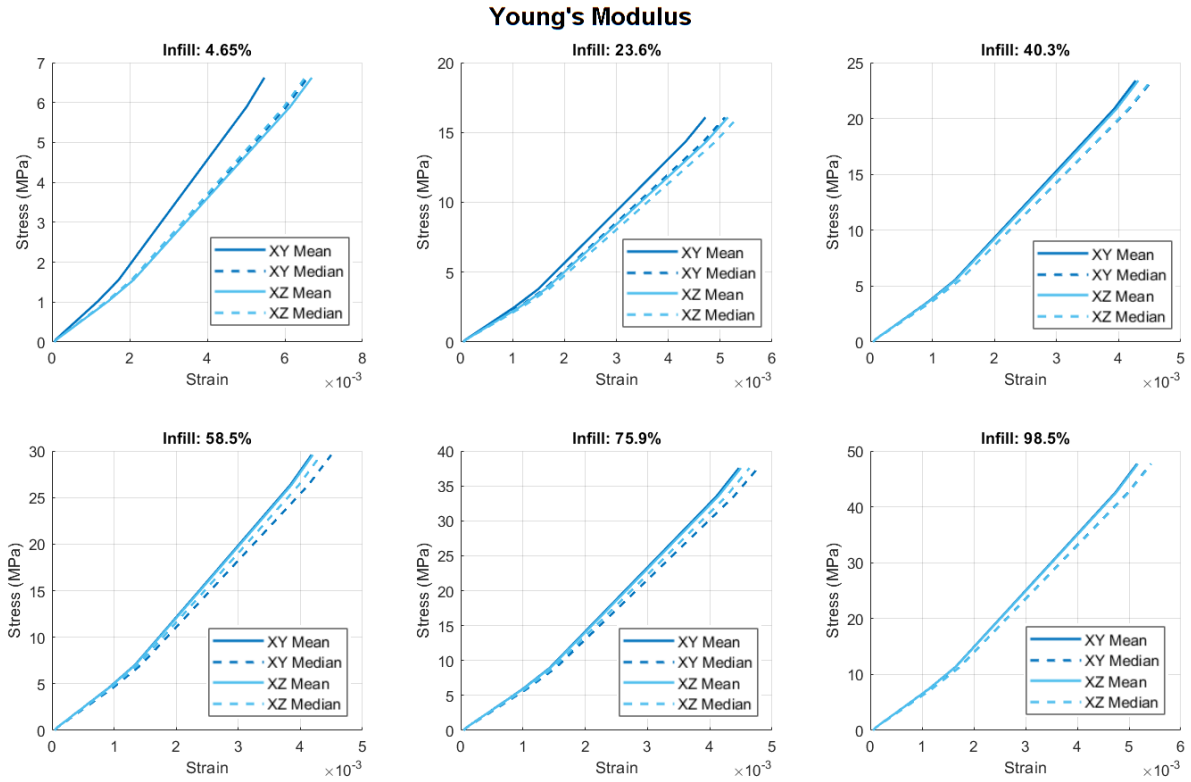


Fig. A.9 Young's moduli at different infills

A.2 Planet Gears Stress Analysis

After the 5:1 gear ratio with the maximum motor torque of 4.6195 N m is 23.1 N m

$$F_t = \frac{23.1\,\text{N m}}{13.95\,\text{mm}} = 1655.91\,\text{N} \tag{A.1}$$

Load sharing across 2 teeth and 3 planet gears:

$$F_{\text{effective}} = \frac{F_t}{2\times 3} = \frac{1655.91\,\text{N}}{2\times 3} \approx 275.99\,\text{N}. \tag{A.2}$$

Substituting numerical values in the Lewis formula from Equation 2.3,

$$\sigma = \frac{275.99\,\text{N}}{18\,\text{mm}\times 1.8\,\text{mm}\times 0.3}. \tag{A.3}$$

Computing the denominator:

$$b\cdot m\cdot Y = 18\times 1.8\times 0.3 \approx 9.72\,\text{mm}^2. \tag{A.4}$$

Hence,

$$\sigma \approx \frac{275.99\,\text{N}}{9.72\,\text{mm}^2} \approx 28.39\,\text{MPa}. \tag{A.5}$$

The UTS (σ_{UTS}) for PA 12 is 53 MPa, leading to Equation A.6:

$$\text{Safety Factor} = \frac{\sigma_{UTS}}{\sigma} = \frac{53\,\text{MPa}}{28.39\,\text{MPa}} = 1.867. \tag{A.6}$$

A.3 Bill of Materials

Image	Alternative Part Name	Count	Part Number	Price
	Eaglepower 8318 120KV Rotor brushless motor	1	KV120	£ 61.79
	61815-2RS Deep Groove Ball Bearing 75x95x10mm	1	SKU: 61815-2RS-ZEN	£ 14.61
	M3 Threaded Insert (Brass)	30	RS PRO: 278-534	£ 2.40
	M5x36 steel dowel pin	4	HDP-5-32-A1	£ 2.80
	625ZZ bearing	3	625ZZ	£ 9.04
	M2.5x4 hex cap screw	6	SSCF-M2.5-4-A2	£ 1.08
	M3x12 countersunk socket head screw	26	SSK-M3-12-A2	£ 2.24
	M3x30 countersunk socket head screw	4	SSK-M3-30-A2	£ 1.26
	M3x16 countersunk socket head screw	6	SSK-M4-16-A2	£ 0.96
			Total	£ 96.18

Fig. A.10 Bill of materials

B. MATLAB CODE

B.1 Lattice Mechanical Properties

The LatticeMechanicalProperties.m file automatically generates, meshes and runs through Abaqus several gyroid TPMS lattices. These simulation results are then imported back into Matlab and the Young's modulus and poisson's ratio are estimated in different directions using tracked wall nodes. Snippets of this code, particularly the Abaqus input struct and Von Mises calculations are based on and significantly adapted from the Latticeworks Github documentation availabe at (Vafaee, 2025).

The full 935 line code for this script is available on *GitHub*. A condensed 119 line pseudocode is shown below containing the general logic and flow of the code, simplified using intuitively named mock functions as the full inclusion of the code was not deemed necessary.

```
% Initialise environment
clear all variables;
close all figures;
clc;

% Settings
updateSim = true;               % Flag to recompute Abaqus simulation
overwrite = false;              % Flag to overwrite existing simulated
    files
coarseMesh = false;             % Flag for coarse mesh testing
figureTitles = true;            % Flag to show figure titles
loadSimDataFromMat = false;     % Flag to load data from .mat file if sim
     exists
doSaveFigures = true;           % Flag to save figures

% Change directory to the simulation path
cd('C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\
    Documents\GitHub\LatticeWorks');

% Start CPU logging
system('logman start CPU_Log');

% Define plot settings
cMap = generateColourMap();     % Custom function to generate colour map
fontSize = 15;
markerSize = 20;

% Control parameters
pointSpacing = 0.15;            % Point spacing for mesh generation

% Geometry generation loop
for i = linspace(1.3,-1.3,64) % Loop over isosurface levels
```

```
30
31      nFigures = 0;                % Initialize figure count
32      levelset = i;                % Isosurface level
33
34      % Geometry dimensions
35      x_length = 30;
36      y_length = 10;
37      z_length = 10;
38
39      % Input structure for mesh generation
40      inputStruct = initialiseInputStruct(x_length, y_length, z_length);
41
42      % Prepare save path for results
43      savePath = generateSavePath(i);
44
45      % Check if simulation is already completed
46      if simulationCompleted(savePath) && ~overwrite
47          fprintf('[%.5g] Lattice density already completed, skipping ...\
                n', i);
48          continue;  % Skip to next iteration
49      end
50
51      % Ensure output directory exists
52      createDirectory(savePath);
53
54      % Prepare Abaqus input file names
55      abaqusInpFileName = createAbaqusInputFileName(savePath);
56      abaqusDATFileName = createAbaqusDatFileName(savePath);
57
58      % Material properties
59      resultStruct.material = setMaterialProperties();
60
61      % Generate geometry
62      [S,X,Y,Z] = gradTPMS(inputStruct);
63      [F,V] = isosurface(X,Y,Z,S,levelset);
64      [fc,vc] = isocaps(X,Y,Z,S,levelset, 'above');
65      [f,v,c] = FV_arrange(F,V,fc,vc);  % Join and clean geometry
66
67      % Visualise surface
68      visualiseSurface(f,v,c,figureTitles,fontSize);
69
70      % Remesh geometry
71      optionStruct.pointSpacing = pointSpacing;
72      [F,V] = ggremesh(f,v,optionStruct);
73      resultStruct.F = F;
74      resultStruct.V = V;
75
76      % Tetrahedral meshing
77      performTetGenMeshing(inputStruct, F, V, coarseMesh, optionStruct);
```

```
78
79      % Visualise mesh
80      meshView(meshOutput);
81
82      % Node labels and boundary condition selection
83      C_vertex = labelBoundaryNodes(V, x_length, y_length, z_length);
84      resultStruct.C_vertex = C_vertex;
85
86      % Visualise boundary conditions
87      visualiseBoundaryConditions(Fb, V, C_vertex, markerSize,
            figureTitles, fontSize);
88
89      % Save all figures
90      saveFigures(savePath, doSaveFigures, nFigures);
91
92      % Prepare Abaqus input structure
93      abaqus_spec = prepareAbaqusInputStruct(inputStruct, V, E_youngs,
            v_poisson, bcEncastreList, bcLoadList);
94
95      % Write Abaqus input file
96      if updateSim
97          abaqusStruct2inp(abaqus_spec, abaqusInpFileName);
98          runAbaqusJob(abaqusInpFileNamePart, savePath);
99      end
100
101     % Import and visualise Abaqus results
102     abaqusData = importAbaqusResults(loadSimDataFromMat, saveFile,
            abaqusDATFileName);
103
104     % Fetch element data from results
105     [E_effectiveStress, E_effectiveStrain] = fetchElementData(abaqusData
            , E);
106
107     % Animate deformations
108     animateDeformations(abaqusData, V, fontSize);
109
110     % Calculate and visualise Poisson's ratio
111     calculatePoissonsRatio(abaqusData, resultStruct);
112
113     % Finalize results and save
114     resultStruct = finalizeResults(resultStruct, abaqusData,
            E_effectiveStress, E_effectiveStrain);
115     saveSimulationResults(savePath, resultStruct);
116
117     % Stop CPU logging
118     system('logman stop CPU_Log');
119 end
```

B.2 Additional Plots

AdditionalPlots.m compiles the results from the 64 Abaqus simulations generated from the code in Section B.1 into subplots. Six subplots are sampled and generated, although this can be adaptively edited.

The full 276 line code for this script is available on *GitHub*. A condensed 109 line pseudocode is shown below containing the general logic and flow of the code, simplified using intuitively named mock functions as the full inclusion of the code was not deemed necessary.

```
% Clear environment and set default settings
clear; close all; clc;

% Define plot settings
setPlotSettings();

% Initialise figures and parameters
count = 1;
nCols = 3; nRows = 2; linspace_steps = 64;
figures = createFigures(10);  % Create 10 figures
original_points = linspace(1.3, -1.3, linspace_steps);
selected_idx = round(linspace(1, linspace_steps, nCols * nRows));

% Process selected density values
for density in original_points[selected_idx]
    resultStruct = loadSimulationResults(density);

    % Generate figures for isosurface and displacement
    plotIsoSurface(figures(1), resultStruct);
    plotOrthographicProjection(figures(2), resultStruct);
    plotIsoSurfaceGeneration(figures(3), resultStruct);
    plotGeogramRemeshed(figures(4), resultStruct);
    plotNodeSets(figures(5), resultStruct);
    plotBoundaryConditions(figures(6), resultStruct);
    plotTrackedDisplacement(figures(7), resultStruct.xz, resultStruct.Uz
        , 'Z');
    plotTrackedDisplacement(figures(8), resultStruct.xy, resultStruct.Uy
        , 'Y');
    plotPoissonsRatio(figures(9), resultStruct);
    plotYoungsModulus(figures(10), resultStruct);

    count++;
end

% Save summary figures
saveSummaryFigures('D:\TechnicalReport\LatticeProperties_LinearFinal\!
    Summary');

% Function Definitions
```

```
function setPlotSettings()
    global figureTitles;
    figureTitles = true;
    % Define colour map and styles
end

function figures = createFigures(num)
    for i = 1:num
        figures(i) = cFigure();
    end
end

function resultStruct = loadSimulationResults(density)
    folderPath = 'D:\TechnicalReport\LatticeProperties_LinearFinal';
    savePath = fullfile(folderPath, sprintf('%.5g_Lattice_Density',
       density));
    matlabPath = fullfile(savePath, 'simulation_results.mat');
    load(matlabPath);
end

function plotIsoSurface(fig, resultStruct)
    figure(fig);
    subplot(nRows, nCols, count);
    title(sprintf('Isovalue: %.3g', resultStruct.level_set));
    hp1 = gpatch(resultStruct.f, resultStruct.v, resultStruct.c, 'none',
        1);
    hp1.FaceColor = 'flat';
    colormap(gca, gjet(6));
    axisGeom(gca, fontSize); camlight headlight;
end

function plotOrthographicProjection(fig, resultStruct)
    figure(fig);
    subplot(nRows, nCols, count);
    title(sprintf('Isovalue: %.3g', resultStruct.level_set));
    hp2 = gpatch(resultStruct.f, resultStruct.v, 'none', 1);
    hp2.FaceColor = [0 0.4470 0.7410];
    axis equal; xlim([min(resultStruct.v(:,1)), max(resultStruct.v(:,1))
       ]);
    ylim([min(resultStruct.v(:,2)), max(resultStruct.v(:,2))]);
end

function plotTrackedDisplacement(fig, xData, yData, direction)
    figure(fig);
    subplot(nRows, nCols, count);
    title(sprintf('Infill: %.3g%%', resultStruct.infill_percentage));
    scatter(xData, yData, 'x', 'MarkerEdgeColor', direction == 'Z' ?
       blue : cyan);
    xlabel('Global X Coordinate (mm)');
```

```
    ylabel(sprintf('%s Displacement (mm)', direction));
    grid on;
end

function plotPoissonsRatio(fig, resultStruct)
    figure(fig);
    subplot(nRows, nCols, count);
    title(sprintf('Infill: %.3g%%', resultStruct.infill_percentage));
    plot(resultStruct.U_x, resultStruct.poisson.poisson_xz_mean, '-', '
        Color', blue);
    plot(resultStruct.U_x, resultStruct.poisson.poisson_xy_mean, '-', '
        Color', cyan);
    xlabel('Applied X Displacement (mm)');
    ylabel('Poisson Ratio');
end

function plotYoungsModulus(fig, resultStruct)
    figure(fig);
    subplot(nRows, nCols, count);
    title(sprintf('Infill: %.3g%%', resultStruct.infill_percentage));
    plot(abs(resultStruct.strain.strain_y_mean), resultStruct.stress, '-
        ', 'Color', blue);
    plot(abs(resultStruct.strain.strain_z_mean), resultStruct.stress, '-
        ', 'Color', cyan);
    xlabel('Strain');
    ylabel('Stress (MPa)');
end

function saveSummaryFigures(folderPath)
    mkdir(folderPath);
    saveFigures(folderPath, true, 0);
end
```

B.3 Summary Struct and Plots

The summaryStructCreation.m merges the simulation_results.mat resultStruct generated from the code in Section B.1 into a single Matlab struct, necessary to plot poisson's ratios and Young's moduli at all timesteps.

The full 349 line code for this script is available on *GitHub*. A condensed 87 line pseudocode is shown below containing the general logic and flow of the code, simplified using intuitively named mock functions as the full inclusion of the code was not deemed necessary.

```
% Clear environment and close all figures
clear all; close all; clc;

% Set default folder for results
defaultFolder = 'D:\TechnicalReport\LatticeProperties_LinearFinal';
```

```
matPath = fullfile(defaultFolder, '!Summary', '
    simulation_results_summary.mat');

% Check if summary results file exists
if fileExists(matPath)
    load(matPath);
else
    % Initialise variables for processing
    count = 1;
    linspace_steps = 64;
    nPoints = 64;
    original_points = linspace(1.3, -1.3, linspace_steps);
    selected_idx = round(linspace(1, linspace_steps, nPoints));
    exclusions = {'f', 'v', 'c', 'F', 'V', 'meshOutput', 'abaqusData', '
        E_effectiveStrain', 'E_effectiveStress'};

    % Loop through selected points
    for i = original_points(selected_idx)
        savePath = fullfile(defaultFolder, sprintf('%.5g_Lattice_Density
            ', i));
        load(matlabPath);

        % Get relevant fields, excluding specified ones
        expectedFields = setdiff(fieldnames(resultStruct), exclusions);

        % Preallocate summary structure on first iteration
        if count == 1
            template = cell2struct(cell(size(expectedFields)),
                expectedFields, 1);
            summaryStruct = repmat(template, 1, linspace_steps);
        end

        % Populate summary structure with existing fields
        for field in expectedFields
            summaryStruct(count).(field) = getFieldValue(resultStruct,
                field);
        end

        disp(count);
        count++;
    end

    % Save summary results
    save(matPath, 'summaryStruct', '-v7.3'); % Use version for large
        sizes
end

% Define colours for plots
blue = [0, 0.4470, 0.7410];
```

```
cyan = [0.3010, 0.7450, 0.9330];

% ---- Plotting Section ----

% Figure 1: Poisson's Ratio
plotPoissonsRatio(summaryStruct, blue, cyan);

% Figure 2: Young's Modulus Fit
fitYoungsModulus(summaryStruct, blue, cyan);

% Figure 3: Young's Moduli R^2 Fit
plotYoungsModuliR2(summaryStruct, blue, cyan);

% Figure 4: Distorted Elements
plotDistortedElements(summaryStruct, blue);

% Figure 5: Infill Percentage vs Level-Set
plotLevelSet(summaryStruct, blue);

% Figure 6: Simulation Time
plotSimulationTime(summaryStruct, blue);

% Figure 7: CPU Load Moving Average
plotCPULoadMovingAverage();

% Figure 8: Number of Tracked Nodes
plotTrackedNodes(summaryStruct, blue, cyan);

% Figure 9: Stress at Each Infill Percentage
plotStress(summaryStruct, blue);

% Figure 10: Poisson's Ratio with Uncertainty
plotPoissonsRatioWithUncertainty(summaryStruct, blue, cyan);

% Figure 11: Young's Moduli with Uncertainty
plotYoungsModuliWithUncertainty(summaryStruct, blue, cyan);

% Save figures in summary folder
saveFiguresInSummaryFolder(defaultFolder);
```

B.4 Topology Optimisation from Matlab Stress FEA

TopOpt.m performs an FEA simulation in Matlab then resamples the data into a grid resulting in a pixelated gear outline. Although it is rasterised the grid dimensions by which this is done can be refined. The stress distribution is created to directly result in a density distribution which directly affects the TPMS.

The full 290 line code for this script is available on *GitHub*. A condensed 107 line pseudocode is shown below containing the general logic and flow of the code, simplified using intuitively named mock functions as the full inclusion of the code was not deemed necessary.

```
% Set working directory and save path
cd("C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\
    Documents\GitHub\LatticeWorks\TopOpt");
savePath = 'D:\TechnicalReport\Variable-TPMS-Figures\TopOpt';

% Load model data
load('sample.mat');
model = createpde("structural", "static-solid");
importGeometry(model, 'Planet-18-Teeth-1.8mm-Mod.step');
scale(model.Geometry, 1e3);  % Scale to mm

% Set material properties
setMaterialProperties(model, 50e6, 1666e6, 0.38);  % (Yield strength, E,
    v)

% Visualise geometry
visualiseGeometry(model);

% Define parameters for teeth and load
teeth_faces =
    [14,24,31,38,48,55,62,72,79,86,96,103,110,120,127,134,144,151];
pressure = calculatePressure(10, 15e-3, teeth_faces);

% Apply loads and constraints
applyBoundaryConditions(model, teeth_faces, pressure);

% Generate mesh and solve
generateMesh(model, 'GeometricOrder', 'quadratic', 'Hmax', 0.0005);
result = solve(model);

% Display results
displayResults(result);

% Interpolate stress data
[xq, yq, zq, stressGrid] = interpolateStressData(result);

% Mask and visualise stress distribution
maskAndVisualiseStress(xq, yq, zq, stressGrid);

% Create and visualise TPMS
createAndVisualiseTPMS(xq, yq, zq, stressGrid, savePath);

% Save figures
saveFigures(savePath, true, 0);
```

```
42
43 % Function Definitions
44 function setMaterialProperties(model, yield_strength, E, v)
45     % Sets the material properties for the model
46     structuralProperties(model, "YoungsModulus", E, "PoissonsRatio", v);
47 end
48
49 function visualiseGeometry(model)
50     cFigure();
51     pdegplot(model, 'FaceLabels', 'on', 'FaceAlpha', 0.5);
52     title('Imported Gear Geometry');
53 end
54
55 function pressure = calculatePressure(torque, teeth_distance,
       teeth_faces)
56     teeth_force = (torque / teeth_distance) / numel(teeth_faces);  % N
57     area_mm2 = 66.441 * 1e-6;  % mm^2
58     pressure = teeth_force / area_mm2;  % Pa
59 end
60
61 function applyBoundaryConditions(model, teeth_faces, pressure)
62     structuralBoundaryLoad(model, "Face", teeth_faces, "Pressure",
           pressure);
63     structuralBC(model, "Face", 8, "Constraint", "fixed");
64 end
65
66 function displayResults(result)
67     fprintf('Max Von Mises: %g MPa.\n', max(result.VonMisesStress) / 1e6
           );
68 end
69
70 function [xq, yq, zq, stressGrid] = interpolateStressData(result)
71     % Interpolates Von Mises stress over a grid
72     x = result.Mesh.Nodes(1, :);
73     y = result.Mesh.Nodes(2, :);
74     z = result.Mesh.Nodes(3, :);
75     vonMisesStress = result.VonMisesStress;
76
77     n = 200;
78     [Xq, Yq, Zq] = meshgrid(linspace(min(x), max(x), n), linspace(min(y)
           , max(y), n), linspace(min(z), max(z), round((max(z) - min(z))/(
           max(x)-min(x))/(n-1)) + 1));
79     F = scatteredInterpolant(x', y', z', vonMisesStress, 'linear', 'none
           ');
80     stressGrid = F(Xq, Yq, Zq);
81     stressGrid(isnan(stressGrid)) = 0;
82 end
83
84 function maskAndVisualiseStress(xq, yq, zq, stressGrid)
```

```
    % Apply masking and visualisation of interpolated stress
    alphaVal = 1;  % Threshold
    shp = alphaShape(xq', yq', z', alphaVal);
    insideMask = inShape(shp, xq, yq, zq);
    stressGrid(~insideMask) = NaN;

    % Create surface plot
    cFigure();
    surf(xq(:,:,1), yq(:,:,1), stressGrid(:,:,1), 'EdgeColor', 'none');
    colorbar();
    xlabel('X Coordinate');
    ylabel('Y Coordinate');
    zlabel('Von Mises Stress');
end

function createAndVisualiseTPMS(xq, yq, zq, stressGrid, savePath)
    % Generate and visualise TPMS structure
    % Logic for TPMS generation and visualisation goes here

    % Save generated mesh
    TR = triangulation(Fsn, Vsn);
    stlwrite(TR, fullfile(savePath, 'OutputGear.stl'));
end
```

B.5 Homogenisation Plots

Plots results from lattice homogenisation using nTopology which returns a full 6x6 stiffness matrix as shown in Equation 3.2. This is computed for the same 64 points used in Section B.1. The custom nTopology file outputted singular .csv's of stiffness matrices and needed to be compiled and visualised.

The full 290 line code for this script is available on *GitHub*. A condensed 33 line pseudocode is shown below containing the general logic and flow of the code, simplified using intuitively named mock functions as the full inclusion of the code was not deemed necessary.

```
% Set folder path and retrieve CSV files
folderPath = 'D:\TechnicalReport\nTop Homogenisation';
files = getCsvFiles(folderPath);  % Function to get .csv files
fileNames = {files.name};

% Define colour schemes
defineColours();

% Extract numeric values from file names and sort
numericValues = extractNumericValues(fileNames);
sortedFilenames = sortFilenames(fileNames, numericValues);

% Read and store data from CSV files
```

```
14 dataTables = loadDataFromFiles(folderPath, sortedFilenames,
       numericValues);
15
16 % Extract infill densities and stiffness matrices
17 [infillPercentage, stiffnessData] = extractInfillAndStiffness(dataTables
       );
18
19 % Plot stiffness matrix heatmaps
20 plotStiffnessHeatmaps(infillPercentage, stiffnessData, sortedFilenames);
21
22 % Perform exponential regression on stiffness components
23 fitResults = performExponentialRegression(infillPercentage,
       stiffnessData);
24
25 % Compare MATLAB results with nTopology results
26 results = compareResults(stiffnessData);
27
28 % Convert results to table and plot
29 resultsTable = convertResultsToTable(results);
30 plotOrthotropicProperties(infillPercentage, resultsTable);
31
32 % Save figures in specified folder
33 saveFigures(folderPath, true, 0);
```

C. MATLAB FUNCTIONS

An overview of the custom functions or external scripts created for this project. Internal functions from the Gibbon and LatticeWorks Matlab libraries will not be included here but are illustrated in their own documentation found at Moerman (2018) and Vafaeefar et al. (2025) respectively.

C.1 Process Log Files Function

```
cd('C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\Documents\GitHub\LatticeWorks');
checkSubfolders = true;

saveFolder = 'C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\Documents\GitHub\LatticeWorks\LogsActual';
masterFile = fullfile(saveFolder, 'master.csv');

logsDir = 'C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\Documents\GitHub\LatticeWorks\LogsActual';

% ensure the Logs directory exists
if ~isfolder(logsDir)
    error('The Logs directory does not exist.');
end

% get a list of all subfolders in the Logs directory
subfolders = dir(logsDir);
subfolders = subfolders([subfolders.isdir]); % keep only directories
subfolders = subfolders(~ismember({subfolders.name}, {'.', '..'})); % exclude . and ..

if isempty(subfolders)
    checkSubfolders = false;
end

% find the most recently modified subfolder
[~, sortIdx] = sort([subfolders.datenum]); % Sort by creation time
subfolders = subfolders(sortIdx);

if checkSubfolders % creating the csv files from the blg files
    for i = 1:length(subfolders)
        recentSubfolder = subfolders(i).name;
        recentSubfolderPath = fullfile(logsDir, recentSubfolder);

        % locate the .blg file in the most recent subfolder
        blgFiles = dir(fullfile(recentSubfolderPath, '*.blg'));

        % check if a .blg file exists
```

```
36          if isempty(blgFiles)
37              error('No .blg file found in the most recent subfolder,
                    check folder permissions. (Admin only by default)');
38          end
39
40          % assume there's only one .blg file; if multiple, process the
                first one
41          blgFile = fullfile(recentSubfolderPath, blgFiles(1).name);
42
43          % define the output .csv file path
44          csvName = sprintf('%.0f', posixtime(datetime('now', 'TimeZone',
                'UTC')));
45          csvFile = fullfile(logsDir, [csvName, '.csv']);
46
47          % convert .blg to .csv using the relog command
48          relogCmd = sprintf('relog "%s" -f csv -o "%s"', blgFile, csvFile
                );
49          system(relogCmd);
50
51          pause(1);
52
53          try
54              rmdir(recentSubfolderPath, 's');
55          catch
56              warning('Failed to delete the folder: %s\nError: %s',
                    recentSubfolderPath, ME.message);
57          end
58      end
59  end
60
61
62  % get list of CSV files sorted by creation date
63  fileList = dir(fullfile(saveFolder, '*.csv'));
64  fileList = fileList(~strcmp({fileList.name}, 'master.csv'));
65  [~, sortIdx] = sort([fileList.datenum]); % Sort by creation time
66  fileList = fileList(sortIdx);
67
68  % loop through each file in order of creation and append to master.csv
69  for i = 1:length(fileList)
70      filePath = fullfile(saveFolder, fileList(i).name);
71
72      % Read the CSV file
73      data = readtable(filePath);
74
75      colIdx = contains(data.Properties.VariableNames, 'Total') & contains
            (data.Properties.VariableNames, 'ProcessorTime') | contains(data.
            Properties.VariableNames, 'GMTStandardTime');
76
77      columnName = data.Properties.VariableNames(colIdx);
```

```
    data = data(:, columnName);

    data.Properties.VariableNames(1) = "Time";
    data.Properties.VariableNames(2) = "Processor Time %";

    numericVars = varfun(@isnumeric, data, 'OutputFormat', 'uniform');
    data(:, numericVars) = fillmissing(data(:, numericVars), 'constant',
        0);

    % append to master.csv (create if it doesn't exist)
    if exist(masterFile, 'file')
        writetable(data, masterFile, 'WriteMode', 'append');
    else
        writetable(data, masterFile);
    end

    delete(filePath);
end

disp('Processing complete. All files merged and deleted.');
```

C.2 Save Figures Function

```
cd('C:\Users\rusco\OneDrive - University of Warwick\Admin\Archives\
    Documents\GitHub\LatticeWorks');
checkSubfolders = true;

saveFolder = 'C:\Users\rusco\OneDrive - University of Warwick\Admin\
    Archives\Documents\GitHub\LatticeWorks\LogsActual';
masterFile = fullfile(saveFolder, 'master.csv');

logsDir = 'C:\Users\rusco\OneDrive - University of Warwick\Admin\
    Archives\Documents\GitHub\LatticeWorks\LogsActual';

% ensure the Logs directory exists
if ~isfolder(logsDir)
    error('The Logs directory does not exist.');
end

% get a list of all subfolders in the Logs directory
subfolders = dir(logsDir);
subfolders = subfolders([subfolders.isdir]); % keep only directories
subfolders = subfolders(~ismember({subfolders.name}, {'.', '..'})); %
    exclude . and ..

if isempty(subfolders)
    checkSubfolders = false;
end

```

```
% find the most recently modified subfolder
[~, sortIdx] = sort([subfolders.datenum]); % Sort by creation time
subfolders = subfolders(sortIdx);

if checkSubfolders % creating the csv files from the blg files
    for i = 1:length(subfolders)
        recentSubfolder = subfolders(i).name;
        recentSubfolderPath = fullfile(logsDir, recentSubfolder);

        % locate the .blg file in the most recent subfolder
        blgFiles = dir(fullfile(recentSubfolderPath, '*.blg'));

        % check if a .blg file exists
        if isempty(blgFiles)
            error('No .blg file found in the most recent subfolder,
                check folder permissions. (Admin only by default)');
        end

        % assume there's only one .blg file; if multiple, process the
            first one
        blgFile = fullfile(recentSubfolderPath, blgFiles(1).name);

        % define the output .csv file path
        csvName = sprintf('%.0f', posixtime(datetime('now', 'TimeZone',
            'UTC')));
        csvFile = fullfile(logsDir, [csvName, '.csv']);

        % convert .blg to .csv using the relog command
        relogCmd = sprintf('relog "%s" -f csv -o "%s"', blgFile, csvFile
            );
        system(relogCmd);

        pause(1);

        try
            rmdir(recentSubfolderPath, 's');
        catch
            warning('Failed to delete the folder: %s\nError: %s',
                recentSubfolderPath, ME.message);
        end
    end
end


% get list of CSV files sorted by creation date
fileList = dir(fullfile(saveFolder, '*.csv'));
fileList = fileList(~strcmp({fileList.name}, 'master.csv'));
[~, sortIdx] = sort([fileList.datenum]); % Sort by creation time
fileList = fileList(sortIdx);
```

```
67
68 % loop through each file in order of creation and append to master.csv
69 for i = 1:length(fileList)
70     filePath = fullfile(saveFolder, fileList(i).name);
71
72     % Read the CSV file
73     data = readtable(filePath);
74
75     colIdx = contains(data.Properties.VariableNames, 'Total') & contains
           (data.Properties.VariableNames, 'ProcessorTime') | contains(data.
           Properties.VariableNames, 'GMTStandardTime');
76
77     columnName = data.Properties.VariableNames(colIdx);
78     data = data(:, columnName);
79
80     data.Properties.VariableNames(1) = "Time";
81     data.Properties.VariableNames(2) = "Processor Time %";
82
83     numericVars = varfun(@isnumeric, data, 'OutputFormat', 'uniform');
84     data(:, numericVars) = fillmissing(data(:, numericVars), 'constant',
           0);
85
86     % append to master.csv (create if it doesn't exist)
87     if exist(masterFile, 'file')
88         writetable(data, masterFile, 'WriteMode', 'append');
89     else
90         writetable(data, masterFile);
91     end
92
93     delete(filePath);
94 end
95
96 disp('Processing complete. All files merged and deleted.');
```

C.3 Simulation Completed Function

```
1 function isCompleted = simulationCompleted(parentDir)
2 % simulationCompleted  Check if a given simulation directory has been
      completed.
3 %   isCompleted = simulationCompleted(parentDir) returns true if the
      base
4 %   directory (parentDir) contains all the required simulation files and
5 %   if the Figures subfolder contains all the required figure files.
6
7 % required files in main folder
8 requiredSimulationFiles = {
9     'Lattice_FEA.com', 'Lattice_FEA.dat', 'Lattice_FEA.inp', '
          Lattice_FEA.msg', ...
```

```
10      'Lattice_FEA.odb', 'Lattice_FEA.prt', 'Lattice_FEA.sta', '
            simulation_results.mat'
11      };
12
13  % required files in Figures subfolder
14  requiredFigureFiles = {
15      'Figure_1.fig', 'Figure_1.png', 'Figure_2.fig', 'Figure_2.png', ...
16      'Figure_3.fig', 'Figure_3.png', 'Figure_4.fig', 'Figure_4.png', ...
17      'Figure_5.fig', 'Figure_5.png', 'Figure_6.fig', 'Figure_6.png', ...
18      'Figure_7.fig', 'Figure_7.png', 'Figure_8.fig', 'Figure_8.png', ...
19      'Figure_9.fig', 'Figure_9.png', 'Figure_10.fig', 'Figure_10.png',
            ...
20      'Figure_11.fig', 'Figure_11.png', 'Figure_12.fig', 'Figure_12.png',
            ...
21      'Figure_13.fig', 'Figure_13.png', 'Figure_14.fig', 'Figure_14.png',
            ...
22      'Figure_15.fig', 'Figure_15.png', ...
23      };
24
25  % check if the base directory exists
26  if ~isfolder(parentDir)
27      isCompleted = 0;
28      return;
29  end
30
31  % check if all required simulation files exist in the base directory.
32  simulationFilesExist = all(cellfun(@(f) exist(fullfile(parentDir, f), '
        file') == 2, requiredSimulationFiles));
33
34  % check if folder exists
35  figuresDir = fullfile(parentDir, 'Figures');
36  if ~isfolder(figuresDir)
37      figureFilesExist = false;
38  else
39      figureFilesExist = all(cellfun(@(f) exist(fullfile(figuresDir, f), '
            file') == 2, requiredFigureFiles));
40  end
41
42  isCompleted = simulationFilesExist && figureFilesExist;
43  end
```

D. ENGINEERING DRAWINGS

D.1 Labelled Assembly

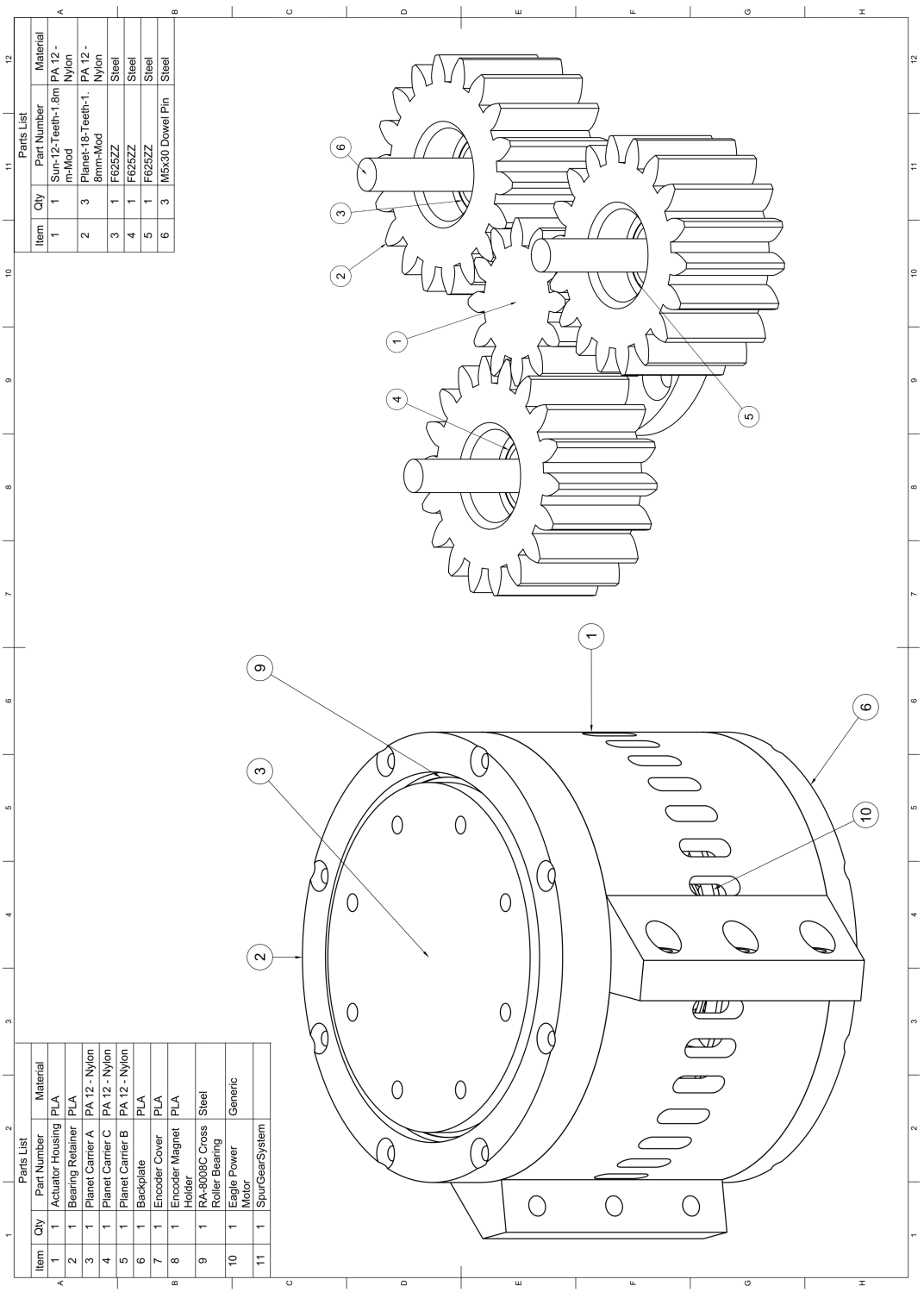


Fig. D.1 Labelled technical drawing for the full assembly and gear assembly

D.2 Technical Drawing

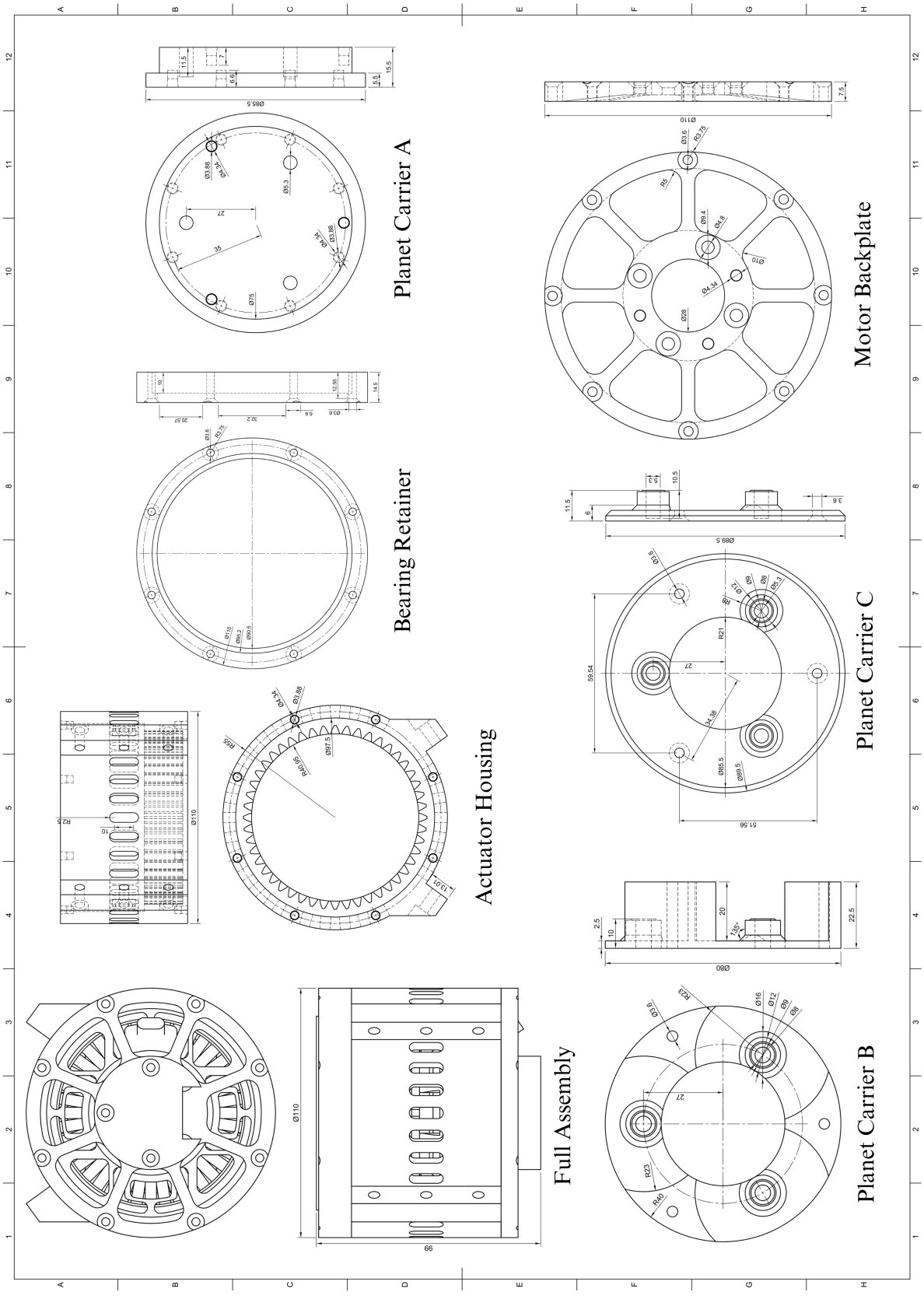
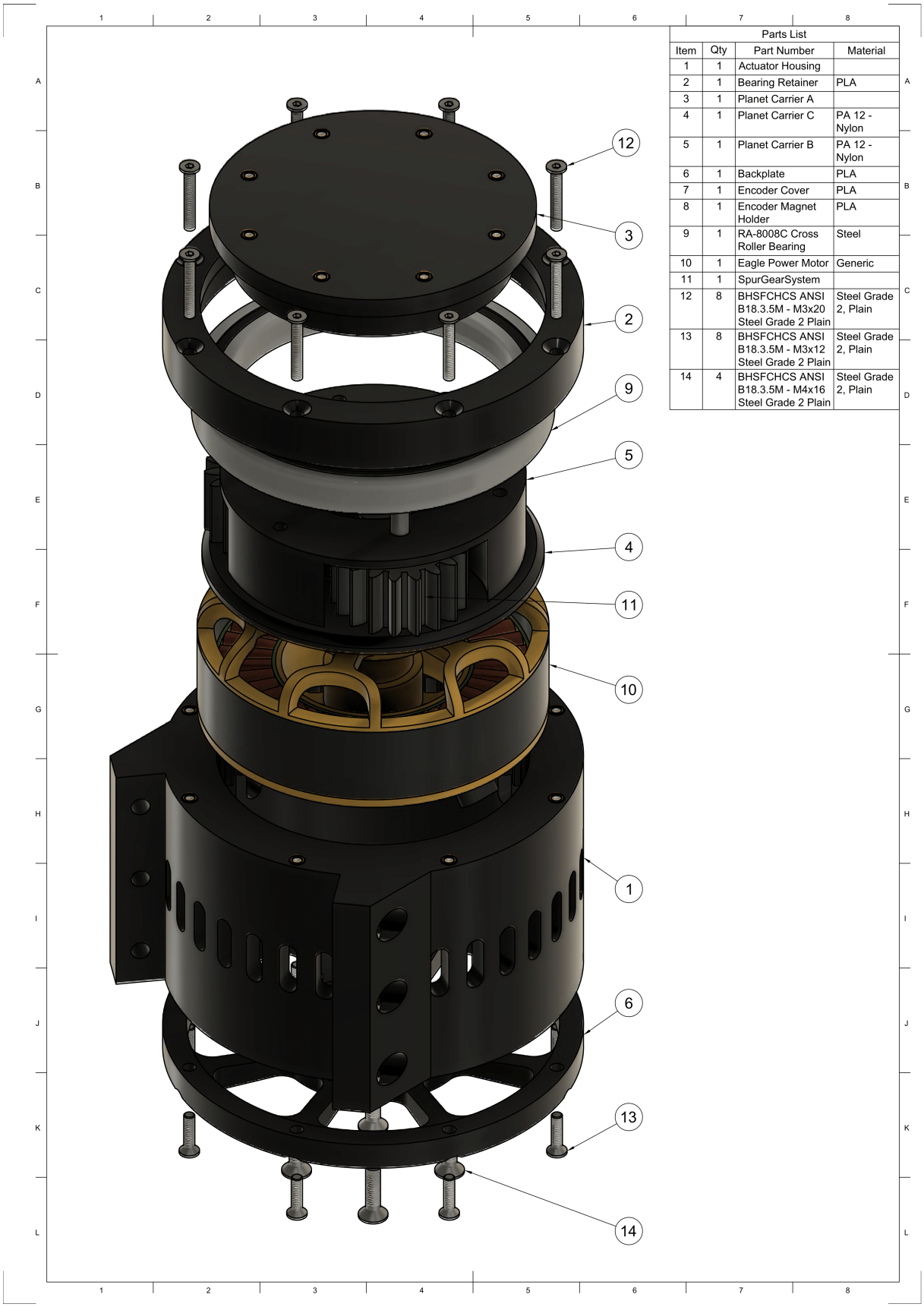


Fig. D.2 Technical drawings of 3D printed actuator components

D.3 Exploded Component View



Parts List

Item	Qty	Part Number	Material
1	1	Actuator Housing	
2	1	Bearing Retainer	PLA
3	1	Planet Carrier A	
4	1	Planet Carrier C	PA 12 - Nylon
5	1	Planet Carrier B	PA 12 - Nylon
6	1	Backplate	PLA
7	1	Encoder Cover	PLA
8	1	Encoder Magnet Holder	PLA
9	1	RA-8008C Cross Roller Bearing	Steel
10	1	Eagle Power Motor	Generic
11	1	SpurGearSystem	
12	8	BHSFCHCS ANSI B18.3.5M - M3x20 Steel Grade 2 Plain	Steel Grade 2, Plain
13	8	BHSFCHCS ANSI B18.3.5M - M3x12 Steel Grade 2 Plain	Steel Grade 2, Plain
14	4	BHSFCHCS ANSI B18.3.5M - M4x16 Steel Grade 2 Plain	Steel Grade 2, Plain

Fig. D.3 Exploded view of actuator with fasteners