

UNIVERSITY OF TUSCIA

Department of Economics, Engineering, Society and Enterprise

Ph.D. Program in Engineering for Energy and Environment

XXXVIII Cycle

Stellarator applications of topological optimization

s.s.d. IIND-03/A

Ph.D. Candidate
Simone Del Nero

Supervisor
Prof. Pierluigi Fanelli

Ph.D. Coordinator
Prof. Andrea Luigi Facci

Co-Supervisor
Victor Prost

A.Y. 2024/2025

Stellarator applications of topological optimization

Simone Del Nero

Doctoral dissertation submitted as part of the final exams for the Ph.D. program in Engineering for Energy and Environment at the University of Tuscia, Department of Economics, Engineering, Society and Business Organization

Keywords: Nuclear Fusion, Stellarator, High Magnetic Field, Structural Design, Topology Optimization

Abstract

The global effort to decarbonize energy production has placed growing emphasis on the development of clean and sustainable alternatives to fossil fuels. Among these, nuclear fusion stands out as a promising long-term solution, offering the potential for virtually limitless and carbon-free energy. Magnetic confinement devices, such as tokamaks and stellarators, represent the most advanced pathway toward practical fusion power. In these systems, powerful magnetic fields are used to confine the plasma, preventing it from coming into contact with the reactor walls and enabling the conditions required for sustained fusion reactions.

To achieve economically viable fusion power, reactor designs must operate at high energy gain (Q), which is strongly correlated with either increasing the reactor size, as in the International Thermonuclear Experimental Reactor (ITER), or increasing the magnetic field strength in the confinement system. Most current research efforts are focused on the second approach, which is driving the development of high-field superconducting magnets capable of sustaining extreme mechanical loads and irradiation damages while producing the magnetic fields required to confine the plasma.

In particular, High-Temperature Superconductors are enabling the design of more compact and efficient magnetic systems, making them an essential technology for future fusion devices. However, the intense magnetic fields produced by such systems give rise to large electro-magnetic forces, leading to significant stresses and deformations within the structural supports and reinforcements of the magnet.

Efficient structural design is therefore critical to ensure both mechanical integrity and material optimization, particularly when the system must operate under tight geometric constraints and such enormous electro-magnetic loads. In this context, the present work focuses on the structural optimization of fusion magnet reinforcement systems for Renaissance Fusion's stellarator, aiming to minimize mass while satisfying stringent stress, displacement, and manufacturing constraints. By integrating numerical modeling and optimization techniques, such as topology optimization, this work provides insight into design solutions capable of withstanding electro-magnetic loads while reducing structural weight, ensuring acceptable mechanical performances and reducing the time to transition into realistic engineering solutions.

Acknowledgments

This research was produced while attending the Ph.D. program in "Engineering for Energy and Environment" at the University of Tuscia, Cycle XXXVIII, with the support of a scholarship co-financed by the Ministerial Decree no. 352 of 9 th April 2022, based on the NRRP - funded by the European Union - NextGenerationEU - Mission 4 "Education and Research", Component 2 "From Research to Business", Investment 3.3 "Introduction of innovative Ph.D. programs that meet the innovation needs of businesses and promote the hiring of researchers by companies".

This project would not have been possible without the collaboration of the University of Tuscia and Renaissance Fusion, who gave me the opportunity to be part of this fantastic project.

I would like to thank my supervisor Prof. Pierluigi Fanelli, my co-supervisor Victor Prost, Gopal Dewangan, Francesco A. Volpe and the entire RF team for their continuous feedback, inspiring discussions and insights which enriched this research.

Finally, I would like to thank my family and Sara for their support over the last few years. Without them, none of this would have been possible.

Contents

Acknowledgments **2**

List of Figures **5**

List of Tables **9**

1 Structural Optimization **10**
1.1 Topology Optimization 11
1.2 How Topology Optimization is accounted in FEA 14
1.3 Filtering technique 17
1.4 Sensitivity Analysis 20
1.4.1 The stiffness maximization Problem with volume constraint . . . 20
1.4.2 The stiffness maximization Problem with volume and stress constraints 23
1.4.3 Manufacturing Constraints in Topology Optimization 30
1.5 Updating Design Variables: Optimization Algorithms 38
1.5.1 Method of Moving Asymptotes (MMA) 39
1.5.2 About normalization in optimization problems 43
1.6 PyMapdl Topology Optimization Tool 44
1.6.1 Benchmark problems 47
1.6.2 Conclusions on benchmark problems 56
1.7 Particle Swarm Optimization 57
1.7.1 PSO: Analytical Formulation 57
1.7.2 Constrained PSO 59
1.7.3 Simply supported beam optimization 61

2 Nuclear Fusion: A brief introduction **64**
2.1 Fundamentals of fusion energy 64
2.2 Experimental Devices for Magnetic Confinement 68
2.3 Lawson Criterion 69
2.4 The role of Renaissance Fusion 69
2.5 Topology Optimization in Nuclear Fusion 71

3 Parametric Modeling for 10 T, 8.5 m bore Stellarator magnet reinforcements 73
3.1 Magnet System Overview . . . 74
3.2 FE model . . . 76
3.3 Minimum Vessel Thickness . . . 78
3.4 Poloidally variable thickness . . . 79
3.5 Double-wall shell with internal poloidal ribs . . . 80
3.6 Double-wall shell with internal beam reinforcements . . . 85
3.7 Uncertainty of PSO results . . . 86
3.8 Conclusions . . . 88

4 Topology Optimization of the 10 T, 52 cm bore Stellarator magnet reinforcements 92
4.1 Magnet System Overview . . . 93
4.2 FE model of initial non-optimized structure . . . 95
4.3 Conceptual design of magnet's reinforcements . . . 99
4.3.1 Maximum Stiffness of initial model s.t. volume constraints . . . 101
4.3.2 Maximum Stiffness of optimized model s.t. volume and extrusion constraints . . . 102
4.3.3 Transitioning from TO to Parametric model . . . 107
4.3.4 Proposed design of magnet's reinforcements . . . 108
4.3.5 Assembling procedure . . . 110
4.3.6 FEA of conceptual design . . . 112
4.4 Conclusions . . . 113

5 Conclusions and Future Works 115

Bibliography 118

Appendix 123
.1 HTS Equivalent Mechanical Properties: Homogenization Technique . . . 124
.2 High level calculation for magnet reinforcement segmentation . . . 126
.3 Tapered Fit Interference . . . 130
.4 8-node brick element . . . 136
.4.1 Element Formulation . . . 136
.4.2 B matrix – Some adjustments . . . 142
.4.3 Python implementation of B matrix . . . 143
.4.4 Comparison between ANSYS and in-house Implementation . . . 147
.5 APDL implementation of the PSO algorithm . . . 151

List of Figures

1.1 (a) Sizing Optimization; (b) Shape Optimization; (c) Topology Optimization. 11
1.2 General Topology Optimization workflow. 14
1.3 Solution of 2D beam problem displaying checkerboard patterns [1]. 17
1.4 Example of structured mesh obtained using sweep method. 31
1.5 Example of structured mesh with the base surface (red dot line), $k-th$ extrusion direction and set of elements sharing common extrusion direction ($\mathcal{C}_k$). 33
1.6 Example of column elements that share the same extrusion direction. . . 34
1.7 Sketch of a 2D filter used to impose: (a) minimum length scale and neighbor filtering; (b) maximum size constraint [2]; (c) maximum size constraint [3]. 35
1.8 Flowchart of PyMAPDL Topology Optimization Tool. 46
1.9 3D Cantilever Beam. 48
1.10 (a) In-house TO: $n_{iter.} = 188$, $t = 2.14$ h, $C_{opt} = 40441$ Nm; (b) ANSYS TO: $n_{iter.} = 51$, $t = 9.30$ min. 48
1.11 3D MBB Beam. 49
1.12 (a) In-house TO: $n_{iter.} = 200$, $t = 6.37$ h $C_{opt} = 341$ Nm; (b) ANSYS TO: $n_{iter.} = 29$, $t = 10.39$ min. 49
1.13 3D L-Bracket. 50
1.14 (a) In-house TO: $n_{iter.} = 179$, $t = 1.00$ h, $C_{opt} = 2381$ Nm; (b) ANSYS TO: $n_{iter.} = 36$, $t = 3.39$ min. 51
1.15 Plate FE model: (a) Ω_{dd} and Ω_{ndd} domains; (b) applied loads and displacement constraints. 51
1.16 Plate TO results: (a) element density plot; (b) top view ($\tilde{\rho} > 0.5$); (c) isometric view ($\tilde{\rho} > 0.5$). 52
1.17 Plate TO results with extrusion and max size constraints: (a) element density plot; (b) top view of optimized rib ($\tilde{\rho} > 0.5$); (c) isometric view of optimized rib ($\tilde{\rho} > 0.5$). 52
1.18 Plate TO convergence plot: (a) normalized objective function with no manufacturing constraints; (b) normalized objective function with manufacturing constraints and decreasing m_l strategy. 53

1.19 Cylinder FE model: (a) Ω_{dd} and Ω_{ndd} domains; (b) applied loads and displacement constraints. 54
1.20 Cylinder TO results: (a) element density plot; (b) top view ($\tilde{\rho} > 0.5$); (c) isometric view ($\tilde{\rho} > 0.5$). 54
1.21 Cylinder TO results with extrusion and max size constraints: (a) element density plot; (b) top view of optimized rib ($\tilde{\rho} > 0.5$); (c) isometric view of optimized rib ($\tilde{\rho} > 0.5$). 55
1.22 Cylinder convergence plot: (a) normalized objective function with no manufacturing constraints; (b) normalized objective function with manufacturing constraints and decreasing m_l strategy. 56
1.23 Simply supported beam model. 62

2.1 Fusion cross-section vs. temperature for different reactions. The D–T reaction peaks at the lowest temperature [4]. 67
2.2 A Tokamak (on the left) and Stellarator (on the right) representation. . . 69

3.1 (a) Magnet system overview, (b) Top view and main dimensions of one magnet. 75
3.2 EM loads plot [MPa]. 76
3.3 Region of stress evaluation (0.5 m from both magnet ends). 77
3.4 Cross-section of the poloidally variable thickness model. 79
3.5 Representation of double wall with poloidal ribs like racetrack shape. . . 81
3.6 Representation of double wall with poloidal variable ribs model. 81
3.7 Representation of double wall with poloidal ribs and internal beams model. 85
3.8 Normalized objective function *vs* iterations: 3.8a constant thickness; 3.8b-3.8c variable poloidal thickness; 3.8d-3.8e double wall shell with racetrack PRs; 3.8f-3.8g double wall shell with poloidal variable PRs; 3.8h-3.8i double wall shell with internal beams. 88
3.9 Normalized objective function variance across PSO swarm *vs* iterations: 3.8a constant thickness; 3.8b-3.8c variable poloidal thickness; 3.8d-3.8e double wall shell with racetrack PRs; 3.8f-3.8g double wall shell with poloidal variable PRs; 3.8h-3.8i double wall shell with internal beams. . . 89
3.10 Results of parametric model in terms of mass (one sector), maximum global stress and maximum displacement. 90
3.11 Parametric modeling solutions: 3.11a constant thickness; 3.11b-3.11c variable poloidal thickness; 3.11d-3.11e double wall shell with racetrack PRs; 3.11f-3.11g double wall shell with poloidal variable PRs; 3.11h-3.11i double wall shell with internal beams. 91

4.1 (a) Top view of magnet arrangements; (b) Magnet cross-section (not drawn to scale). 95
4.2 Two-dimensional distribution of the density current magnitude [MA/m], with axes normalized w.r.t. the magnet's axial and poloidal directions. . 95
4.3 Two-dimensional distribution of the magnetic field magnitude [T], with axes normalized w.r.t. the magnet's axial and poloidal directions. 96
4.4 Two-dimensional distribution of the Laplace force magnitude [MPa], with axes normalized w.r.t. the magnet's axial and poloidal directions. 96
4.5 Structural FE model of magnet (Ω_{ndd}), former (Ω_{ndd}) and design domain (Ω_{dd}). 97
4.6 Displacement field [mm] of initial non-optimized structure. 100
4.7 Stress field [MPa] of initial non-optimized structure. 100
4.8 Equivalent strain field [m/m] of magnet (initial non-optimized structure). 101
4.9 Element density plot of Initial Topology Optimized model with 30% of volume fraction ($m_{final} = 3.20$ t). 102
4.10 Element density plot of Initial Topology Optimized model with 20% of volume fraction ($m_{final} = 2.30$ t). 103
4.11 Element density plot of Initial Topology Optimized model with 10% of volume fraction ($m_{final} = 1.30$ t). 103
4.12 Reconstructed design domain for second-stage optimization. 104
4.13 Extrusion regions. 105
4.14 Topology optimization solutions with volume and extrusion constraints. Figures (a)–(c): 3 layers, volume fractions 50%, 37.5%, 25%; Figures (d)–(f): 6 layers, 45%, 31%, 17%; Figures (g)–(i): 9 layers, 36%, 21%, 5%. 106
4.15 Lateral view of the parametric model for DoE. t_{MR}: Magnet Reinforcement thickness; t_{former}: former thickness; t_{CR}: Central Ring radial thickness; t_{PR}: Poloidal Ribs radial thickness. 107
4.16 Results of parametric model in terms of mass (one sector), maximum global stress and maximum magnet strain. 108
4.17 Results of parametric model in terms of mass (one sector), maximum global stress and maximum magnet displacement. 109
4.18 Representation of the conceptual design of the magnet's reinforcements. . 110
4.19 Exploded view: 1) former, 2) magnet, 3) OC, 4) MR, 5) PRs, 6) former caps, 7) CR vertical plate, 8) CR bottom plate, 9) CR upper plate. . . . 111
4.20 Displacement field of proposed design [mm]. 113
4.21 Stress field of proposed design [MPa]. 113

1 Representation of the segmented reinforcement: (a) schematic cross-section highlighting the separating loads acting on the joint; (b) finite element model of the separated reinforcement showing the location of fixed constraints. 127
2 Simplified top view of the segmented reinforcement joint, illustrating the bolt arrangement, bolt rows, plate radial length, sides, and bolt spacing Δs. 128
3 Cross-section of press-fitted straight cylinders: a : inner radius; b : radius at interference; c : outer radius. 131
4 Cross-section of tapered fit: r: radial coordinate; z : axial coordinate; L : axial length; b : tapered radius; α: tapered angle; b_{min}: minimum tapered radius; b_{max}: maximum tapered radius; p_{f1}: contact pressure on the outer cylinder; p_{f2}: contact pressure on the inner cylinder. 133
5 FE model of two cylinder connected with tapered fit interference. 133
6 Radial stress σ_r [MPa]. 135
7 Tangential stress σ_t [MPa]. 135
8 Pressure contact p_f at the interface [MPa]. 135
9 Isoparametric 8-node brick element: (a) element in the physical space, (b) element in the natural space. 136
10 Isoparametric 8-node brick element: (a) node positions and numbering, (b) element positions and numbering. 147
11 **B**: from 1^{st} to 11^{th} column. 150
12 **B**: from 12^{th} to 23^{rd} column. 150
13 **B**: 24^{th} column. 150

List of Tables

1.1 Beam optimization results for different settings of maximum iterations and particle number using PSO algorithm. 62

3.1 Optimization results for constant-thickness reinforcement model. 78
3.2 Optimization results for poloidally variable thickness configuration. . . . 80
3.3 Optimization results for reinforcement systems with internal PRs. 85
3.4 Optimization results for double-wall shell with internal beam reinforcements. 86

4.1 Key parameters of the magnet. 94
4.2 Equivalent mechanical properties of homogenized HTS material. 98
4.3 Mechanical response of initial model under varying volume constraints. . 102
4.4 Topology optimization results for different extrusion constraints. Here, n_{layers} denotes the number of extrusion layers and f the applied volume fraction. Reported values include final mass m_{final}, maximum magnet displacement $u_{mag.}$, equivalent strain $\varepsilon_{mag.}$, and maximum stress σ_{max}. . . 106

1 Results of bolt number estimation for magnet reinforcement segmentation. 127
2 Parameters used in FE model: f : interference; L : axial length; α: tapered angle; a: minimum inner cylinder radius; c: maximum outer cylinder radius; b_{min}: minimum tapered radius; b_{max}: maximum tapered radius; $E_i = E_o$: Young's modulus of inner and outer cylinders; $\nu_i = \nu_o$: Poissoin's ratio of inner and outer cylinders. 134
3 Comparison between analytical and numerical results for the tapered fit interference case (at the interference radius b): p_f: contact pressure; $\sigma_{r,i}$: radial stress on inner cylinder; $\sigma_{r,o}$: radial stress on outer cylinder; $\sigma_{t,i}$: tangential stress on inner cylinder; $\sigma_{t,o}$: tangential stress on outer cylinder. 135
4 ANSYS node ordering for the SOLID185 element in the (r, s, t) natural coordinate system. 138
5 Gauss integration points and corresponding weighting factors in 1D. For 3D, weights are multiplied: $w_i\, w_j\, w_k$. 142
6 Nodal coordinates of element 1. 148
7 Comparison of results from ANSYS APDL and the in-house code for rlement 1 at the first Gauss point. 150

1 Structural Optimization

In modern engineering, the demand for lighter, stronger, and more efficient structures continues to grow, since it is driven by the pursuit of performance, cost-effectiveness, and sustainability. Whether designing aerospace components, civil infrastructure, or the complex internal architecture of a nuclear fusion reactor, engineers are constantly faced with the challenge of making the best possible use of available materials. Structural optimization provides a systematic approach to this challenge, allowing engineers to mathematically and computationally determine the most efficient configurations of a structure while meeting specific performance and manufacturing criteria.

Rather than relying solely on intuition or iterative trial and error, structural optimization leverages numerical methods to guide the design process. It helps answer fundamental questions such as: *Where should material be placed? What shape should the structure take? How can weight be reduced without compromising strength or functionality?* These questions are particularly critical where structures must withstand extreme forces in constrained spaces, like those in fusion experiments which are using evenly increasing magnetic field for achieving the positive Q (meaning that the power produced by fusion reaction is higher then the injected).

Structural optimization is typically categorized into three main classes, each targeting different aspects of the design process (Fig. 1.1):

- **Sizing Optimization:** involves adjusting parameters such as cross-sectional areas, thickness distributions, or material properties within a fixed structural layout. For example, one might seek the optimal thickness distribution of a linearly elastic plate or the ideal member sizes in a truss structure. The objective typically involves minimizing or maximizing a physical quantity such as compliance (a measure of structural stiffness), maximum stress, or deflection, all while adhering to equilibrium conditions and design constraints. In this context, the design variables (e.g., thickness or cross-sectional area) and the state variables (e.g., stress or displacement) are defined on a fixed domain, which remains unchanged throughout the optimization process.

- **Shape Optimization:** extends this approach by treating the geometry of the domain itself as a variable. Rather than adjusting internal parameters within a fixed boundary, shape optimization seeks the best contour or surface that improves performance criteria, such as reducing stress concentrations or minimizing drag, under the given loading and boundary conditions.
- **Topology Optimization:** it is the most general optimization allowing for the creation or removal of material within the design domain. It determines not only the shape but also the optimal number, location, and distribution of voids or material regions. This method enables the exploration of novel structural configurations, often leading to highly efficient and unconventional geometries that would not be achievable through traditional design approaches.

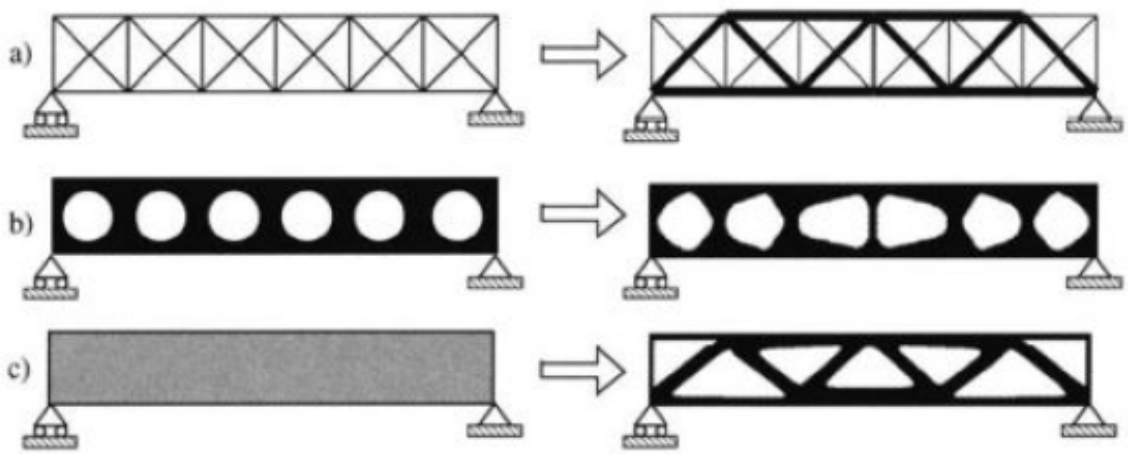


Figure 1.1: (a) Sizing Optimization; (b) Shape Optimization; (c) Topology Optimization.

1.1 Topology Optimization

Topology Optimization (TO) is a powerful computational technique aimed at determining the optimal material layout within a predefined design domain, subject to performance objectives and constraints. Unlike traditional structural design methods, which often rely on manual iterations or fixed geometries, TO enables the automated synthesis of structural forms that maximize performance metrics, such as stiffness or load-bearing capacity, while minimizing material usage. This approach is especially valuable in high-performance and weight-sensitive applications, including aerospace, automotive, and nuclear fusion systems.

The main goal of TO is to distribute material within the design space in the most efficient way possible, often resulting in organic and counterintuitive geometries that

would be difficult to conceive through conventional design approach. The outcomes can significantly improve structural performance, reduce weight, and enable innovative, manufacturable solutions.

Several TO problems were addressed since its inception, and they can be classified according to their primary objectives and constraints. Common problem types include:

- **Stiffness maximization (Compliance minimization) [5]:** minimize the compliance (or maximize the stiffness) of a structure for a given volume of material;
- **Stress-constrained optimization [6]:** minimize the weight or material usage while ensuring that the stress in the structure does not exceed a certain limit;
- **Frequency optimization [7]:** optimize the natural frequencies of a structure to avoid resonance with external vibrations;
- **Multi-objective optimization [8]:** optimize for multiple criteria simultaneously, such as minimizing weight while maximizing stiffness and ensuring a certain level of manufacturability;
- **Thermal optimization [9]:** optimize the thermal performance of a structure, such as maximizing heat dissipation or minimizing thermal stress;
- **Buckling optimization [10]:** minimize the propensity of a structure to buckle under compressive loads;
- **Fluid-structure interaction optimization [11]:** optimize structures that interact with fluid flows, such as minimizing drag or optimizing load-bearing capabilities in fluid environments.

A typical TO workflow, illustrated in Fig. 1.2, involves the following sequential steps:

1. Initialization: construct the Finite Element (FE) model, including the mesh, boundary conditions, and applied loads. Define the design space, Design Variables (DVs), and the objective and constraint functions;
2. Filtering Techniques: apply filtering or regularization methods to DVs. This step is crucial to eliminate numerical instabilities such as mesh dependency, checkerboard patterns, and non-physical local minima;
3. Finite Element Analysis (FEA): perform structural analysis to obtain the response of the system under the applied loads for the current material distribution;

4. Sensitivity Analysis: compute the derivatives of the objective function and constraints with respect to the DVs. These sensitivities quantify the influence of each variable on the performance metrics and guide the optimization process;

5. Sensitivity Filtering: apply a spatial filtering operation to the raw sensitivity values computed in the previous step. This procedure smooths out abrupt local variations and ensures that the sensitivities are consistent with the physical length scale imposed by the filter radius. By redistributing sensitivities across neighboring elements, the method avoids non-physical artifacts such as checkerboard patterns and mesh dependency, while promoting well-defined structural features;

6. Design Variable Update (Optimization): update the DVs using numerical optimization algorithms, such as the Method of Moving Asymptotes (MMA), Optimality Criteria (OC), or other gradient-based solvers. The updated material distribution reflects a better-performing structure according to the objective function;

7. Convergence Check: evaluate convergence based on defined criteria such as maximum number of iterations, change in objective function value, or variation in DVs. If convergence is not achieved, the process returns to the FEA step and continues iteratively.

This iterative loop continues until a satisfactory solution is found, balancing performance objectives with the given constraints. In this context, a satisfactory solution is mathematically defined by meeting a specified tolerance, typically chosen by the user. A looser tolerance can accelerate convergence but may yield less precise results, whereas a tighter tolerance ensures higher accuracy at the cost of increased computational effort and slower convergence.

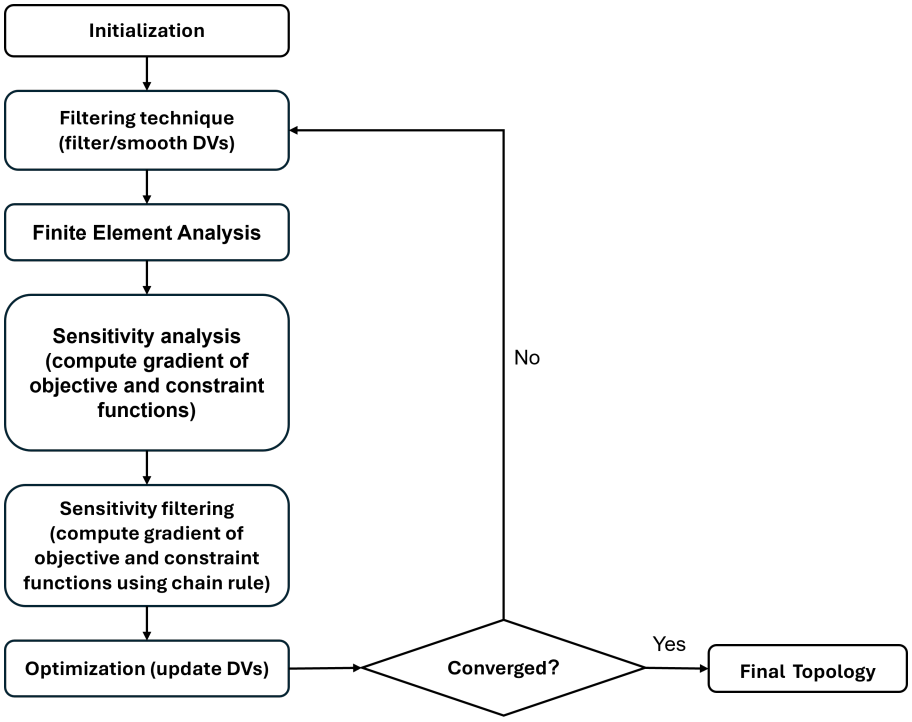


Figure 1.2: General Topology Optimization workflow.

1.2 How Topology Optimization is accounted in FEA

Various material models have been proposed in literature to solve the discrete TO problem with continuous DVs. The most prominent approaches include [12]:

- **Density-based methods:** the Solid Isotropic Material with Penalization (SIMP) approach interpolates material properties based on element density and is widely used for its simplicity and efficiency;
- **Topological derivatives:** these provide a mathematical framework to assess the sensitivity of the objective function to the creation of an infinitesimal hole, guiding material removal;
- **Level-set methods:** the structure's boundary is implicitly represented by a level-set function, allowing smooth evolution of shape and topology;
- **Phase-field methods:** these use a continuous scalar field to represent material distribution, enabling smooth transitions between solid and void regions;
- **Evolutionary approaches (ESO/BESO):** evolutionary Structural Optimization (ESO) and its extension, Bidirectional ESO (BESO), iteratively remove or add material based on element performance.

In this work, the focus is primarily on density-based material models, as they have been employed throughout the study. Their mathematical formulation is relatively simple and straightforward, making them widely adopted in the field and extensively supported by most commercial FE software packages.

As previously mentioned, density-based methods are among the most widely used approaches in TO due to their simplicity and ease of implementation. These methods parameterize the material distribution problem using a scalar field of material densities, which serve as the DVs. In a discretized design domain, the mechanical properties of each finite element (particularly the stiffness tensor) are interpolated between void and solid using a power law function. This penalization strategy is commonly known as the Solid Isotropic Material with Penalization (SIMP) method [13].

The SIMP approach is based on a heuristic relationship between the filtered relative density of an element and its corresponding Young's modulus $E_i(\tilde{\rho}_i)$, which is expressed as:

$$E_i = E_i(\tilde{\rho}_i) = \tilde{\rho}_i{}^p E_0, \quad \tilde{\rho}_i \in (0, 1] \tag{1.1}$$

Here, E_0 is the elastic modulus of the solid material, and p is the penalization power (typically set to $p = 3$), which encourages the optimization to favor values of $\tilde{\rho}_i$ close to 0 or 1, thereby driving the solution toward a black-and-white distribution of material, where $\tilde{\rho}_i$ is the filtered density of the i-th element, representing its relative contribution to resisting the applied loads in the FE model.

To avoid numerical instabilities associated with near-zero stiffness values, especially when $\tilde{\rho}_i$ approaches zero, a modified version of the SIMP method is often employed (referred as MSIMP). This formulation ensures that the stiffness matrix remains well-conditioned by introducing a small, non-zero lower bound E_{min} on the Young's modulus of void elements:

$$E_i = E_i(\tilde{\rho}_i) = E_{min} + \tilde{\rho}_i{}^p (E_0 - E_{min}), \quad \tilde{\rho}_i \in (0, 1] \tag{1.2}$$

In this expression, E_{min} is typically chosen to be a very small fraction of E_0 (e.g., 10^{-9} Pa), sufficient to prevent singularities in the global stiffness matrix while still effectively modeling a void.

For each element i, the original stiffness matrix $\mathbf{k_{i,0}}$, corresponding to the solid material with full density ($\tilde{\rho}_i = 1$), is scaled according to either the SIMP (Eq. 1.1) or MSIMP (Eq. 1.2) power laws. The resulting element stiffness matrix, using the SIMP method, becomes:

$$\mathbf{k_i}(\tilde{\rho}_i) = \tilde{\rho}_i{}^p E_0 \mathbf{k_{i,0}} \tag{1.3}$$

This modification ensures that the mechanical behavior of each element reflects its current material state, ranging from nearly void (low stiffness) to fully solid (high stiffness). The global stiffness matrix $\mathbf{K}(\tilde{\boldsymbol{\rho}})$ of the structure is then assembled in the usual manner by summing the contributions of all element matrices:

$$\mathbf{K}(\tilde{\boldsymbol{\rho}}) = \sum_{i=1}^{e_{tot}} \mathbf{K}_i(\tilde{\rho}_i) \tag{1.4}$$

where e_{tot} is the total number of elements. This formulation couples the density field with the FEA process, allowing the optimization algorithm to iteratively update the DVs vector $\tilde{\boldsymbol{\rho}}$ based on structural performance metrics such as compliance, stress, or displacement.

In the Finite Element Method (FEM) the 3D constitutive matrix for an isotropic element is interpolated from void to solid as:

$$\mathbf{D_i}(\tilde{\rho}_i) = E_i(\tilde{\rho}_i)\mathbf{D_{i,0}} \tag{1.5}$$

where $\mathbf{D_{i,0}}$ is the constitutive matrix with unit Young's modulus, and it is defined as [14]:

$$\mathbf{D_{i,0}} = \frac{1}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2 \end{bmatrix} \tag{1.6}$$

the mechanical parameter ν, in Eq. 1.6, is the Poisson's ratio.

Using the FEM, the stiffness element matrix $\mathbf{k_i}(\tilde{\rho}_i)$ is the volume integral of the elements constitutive matrix $\mathbf{D_i}(\tilde{\rho}_i)$ and the strain–displacement matrix $\mathbf{B}$ in the form of:

$$\mathbf{k_i}(\tilde{\rho}_i) = \int_{V_i} \mathbf{B}_a^T \mathbf{D_i}(\tilde{\rho}_i)\mathbf{B}_a \, dV_i = \tilde{\rho}_i{}^p E_0 \mathbf{k_{i,0}} \tag{1.7}$$

where the dV_i is the $i-th$ elemental volume in the FE model. Therefore, in a static structural analysis with isotropic materials, the element stiffness matrix $\mathbf{k_i}(\tilde{\rho}_i)$ is updated

by modifying the constitutive matrix to reflect the material density. From a finite element perspective, Young's modulus E_i associated with the element $i-th$ within the design domain Ω_{dd} must be updated according to Eq. 1.1. (or Eq. 1.2 if MSIMP is employed) .

Once the global stiffness matrix $\mathbf{K}(\tilde{\rho})$ is updated, it is then used to solve the equilibrium equation with the actual material distribution:

$$\mathbf{K}(\tilde{\boldsymbol{\rho}})\ \mathbf{U}(\tilde{\boldsymbol{\rho}}) = \mathbf{F} \tag{1.8}$$

where Eq. 1.8 is solved within each TO iteration, and consider the current material distribution.

1.3 Filtering technique

Several restriction methods have been suggested in the literature to eliminate the checkerboard effect and mesh-dependency in TO. Many of the proposed restriction methods have the secondary effect of introducing a mesh-independent length scale into the TO problem, thereby reducing, but not eliminating, the effect of mesh dependency. The checkerboard effect (see Fig. 1.3) in TO is a numerical instability that manifests itself as an alternating pattern of filled and empty material regions in the optimized design, creating a non-physical two-color pattern instead of a uniform single-density distribution, i.e. a pattern in which neighbouring elements are assigned opposite densities (e.g. 0 and 1).

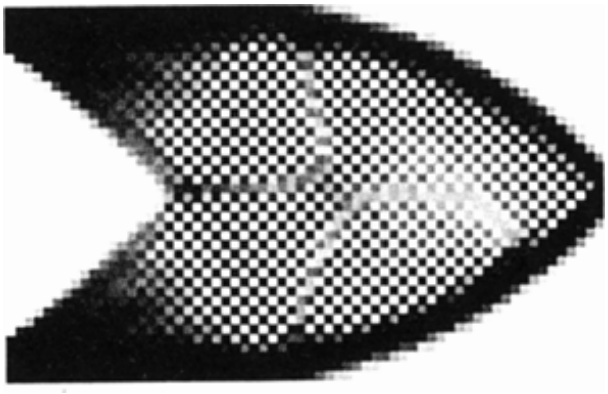


Figure 1.3: Solution of 2D beam problem displaying checkerboard patterns [1].

In general, filtering technique consists of [15]:

- **Density Filtering:** the density filtering technique smooths out the element densities to reduce mesh-dependency and eliminate checkerboard patterns. The idea is to compute a filtered density $\tilde{\rho}_i$ for each element i as a weighted average of the densities ρ_j of the surrounding elements j;

- **Sensitivity Filtering:** the sensitivities of the objective and constraint functions are smoothed instead of the densities. This helps in achieving a similar goal of reducing numerical instabilities and mesh-dependency.

From the implementation point of view, both methods utilize a matrix of weighting values that incorporates the sensitivities or densities of neighbouring elements. The values of elements within the weighting matrix are dependent upon a user-defined filtering radius as well as the distances between element centroids in each finite element discretization (it is strongly suggested to use a filter radius larger then at least two times the minimum element size in the FE model). For centroid-to-centroid distances greater than the user-defined filter radius, the weighting value is zero. The weighting matrix depending only on the geometry of the design domain finite element discretization, it needs to be calculated only once for a given TO problem. For larger mesh model, the process may require a significant amount of time.

Sensitivity filtering works by modifying the objective function f after solving the FE problem; the sensitivity is heuristically modified as weighted averages of the sensitivities in mesh-independent neighborhood as [5]:

$$\frac{\partial \tilde{f}}{\partial \rho_i} = \frac{1}{\max(\rho_{min}, \rho_i) \sum\limits_{j \in \Omega^i_{neighbor}} H_{ij}} \sum_{j \in \Omega^i_{neighbor}} H_{ij} \rho_j \frac{\partial \tilde{f}}{\partial \rho_j} \tag{1.9}$$

In contrast to sensitivity filters, density filters are not heuristic. This means they can be used with line search optimization methods and convergence checks based on the Karush-Kuhn-Tucker (KKT) conditions, as long as the filter and its sensitivity terms are correctly implemented [16].

The process of applying a density filter may be summarized as follows:

1. The un-filtered element-densities $\boldsymbol{\rho}$ are passed to a filtering function F, yielding the filtered element-densities $\tilde{\boldsymbol{\rho}}$;

2. The stiffness element matrices of each element are updated using the filtered element-densities $\tilde{\boldsymbol{\rho}}$, then FEA is performed (in FE softwares, elemental material properties are modified element by element);

3. Objective and constraint functions are calculated using outputs from FEA;

4. The objective and constraint sensitivities w.r.t. the un-filtered element-densities, $\boldsymbol{\rho}$, are calculated. In this step, it is important to account for chain rule terms associated with differentiating the filtering function with respect to the un-filtered element-densities;

5. The objective, constraint, and sensitivities w.r.t. the un-filtered element-densities are used to update the current design point (i.e., the vector $\tilde{\boldsymbol{\rho}}$).

It is important to note that the described process is repeated at each TO iteration.

Several number of different density filters have been studied and proposed in the literature, but the most popular and widely used is the linear density filter [17], which modifies existing element-densities to be linearly weighted based on the element-densities of neighbouring element. In this work, linear density filter were employed, therefore the main focus will be given to describe its mathematical formulation. Regardless of the type of filter used, the neighboring elements of the i-th element are identified based on the distance between their centroids. Specifically, an element e_j is considered a neighbor of element e_i if it lies within a predefined filter radius $r_{\min}$. This neighborhood can be mathematically defined as (see Fig. 1.7a):

$$\Omega^i_{\text{neighbor}} = \left\{ e_j \;\middle|\; \|x_i - x_j\| \leq r_{\min},\ \forall i, j \in \Omega_{\text{dd}} \right\} \tag{1.10}$$

where x_i and x_j represent the coordinates of the centroids of elements e_i and e_j, respectively, and Ω_{dd} denotes the design domain. In vectorized form, the density filtering is obtained by multiplying the filter matrix $\mathbf{H}$ and the element density vector:

$$\tilde{\boldsymbol{\rho}}(\boldsymbol{\rho}) = \mathbf{H}\ \boldsymbol{\rho} \tag{1.11}$$

and the corresponding components of the filtered element density vector are computed as:

$$\tilde{\rho}_i = \frac{\sum\limits_{j \in \Omega^i_{neighbor}} h_{ij} v_j \rho_j}{\sum\limits_{j \in \Omega^i_{neighbor}} h_{ij} v_j} \tag{1.12}$$

In Eq. 1.11, the components of the filter matrix $\mathbf{H}$ are defined as:

$$H_{ij} = \frac{h_{ij} v_j}{\sum\limits_{j \in \Omega^i_{neighbor}} h_{ij} v_j} \tag{1.13}$$

where h_{ij} is the so-called weighting function, which is given by the linearly decaying (cone-shape) function:

$$h_{ij} = r_{min} - dist(i, j), \quad \forall j \in \Omega^i_{neighbor} \tag{1.14}$$

where the operator $dist(i, j)$ is the centroid distance between elements i and j. Other types of weighting functions can be used, like constant or exponential.

1.4 Sensitivity Analysis

Depending on the nature of the problem being solved, different objective and constrains function may be defined. Since most of the effort in TO has been pointed towards gradient based algorithms, due to their numerical stability, when defining the TO problem one needs to evaluate numerically at least first order derivatives of both objective and constraint functions w.r.t. element density vector $\boldsymbol{\rho}$. Gradient free algorithms, like Particle Swarm Optimization (PSO) or Genetic Algorithms (GA), suffer for optimizing a multidimensional problem with more then hundred of variables, and are indeed not widely employed in TO (indeed, the number of DVs are usually equal to the number of elements in the FE model). In the next sections, two of the most important TO problems in classical solid mechanics are presented to give the reader a mathematical formulation that can be easily implemented in a language program like Python or Matlab. In this work the stiffness maximization problem with volume and extrusion constraints were employed due to its ease implementation and faster convergence. Additionally, the maximum size constraint is shown due to its importance when dealing with large scale structure, like future fusion power plants. In this context, for manufacturing reasons might be essential to reduce the size of the structural components.

1.4.1 The stiffness maximization Problem with volume constraint

The easiest TO problem is the stiffness maximization, also well known as minimum compliance optimization. The objective of the minimum compliance problem is to find the material density distribution $\tilde{\boldsymbol{\rho}}$ that minimizes the structure's deformation under the prescribed support and loading condition (or to minimize the work done by the external forces), with a predefined amount of volume (or mass in case of homogeneous material with constant density).

From the mathematical point of view, the minimum compliance problem can be formulated as following:

$$
\begin{aligned}
\min_{\boldsymbol{\rho}} \quad & C(\tilde{\boldsymbol{\rho}}) = \mathbf{U}^T \mathbf{K}(\tilde{\boldsymbol{\rho}}) \mathbf{U} = \sum_{i=1}^{e_{tot}} \tilde{\rho}_i^p E_0 \mathbf{u_i}^T \mathbf{k_{i,0}} \mathbf{u_i} \\
\text{s.t.} \quad & \mathbf{K}(\tilde{\boldsymbol{\rho}}) \mathbf{U} = \mathbf{F} \\
& G_v = \frac{\sum_{i=1}^{e_{dd}} \tilde{\rho}_i \cdot v_i}{f \cdot V_0} - 1 \leq 0 \\
& \rho_{min} \leq \rho_i \leq 1, \quad \forall i \in \Omega_{\text{dd}}
\end{aligned}
\tag{1.15}
$$

where:

- $C(\tilde{\boldsymbol{\rho}})$: total structural compliance;
- $\mathbf{U}$: nodal displacement vector;
- $\mathbf{K}(\tilde{\boldsymbol{\rho}})$: global element stiffness matrix;
- $\mathbf{u_i}$: element nodal displacement vector;
- $\mathbf{k_{i,0}}$: unit element stiffness matrix;
- $\mathbf{F}$: external force vector;
- G_v: global volume constraint:
- v_i: $i-th$ element volume;
- f: volume fraction;
- V_0: total volume of Ω_{dd};
- ρ_{min}: minimum element density.

When coupling TO with commercial FE software, one need to replicate internal FE matrices which might be not ease or impossible to access. Therefore, to avoid the implementation of $\mathbf{k_i}(\tilde{\rho}_i)$, which may differ from the formulation used in the FE software being considered, the strain energy $w_i(\tilde{\rho}_i)$ for each element in the mesh can be used directly to calculate the compliance of the element $c_i(\tilde{\rho}_i)$. This is particularly convenient since strain energy is typically one of the outputs of FEA. Specifically, the element strain energy $w_i(\tilde{\rho}_i)$ is defined as:

$$
w_i(\tilde{\rho}_i) = \frac{1}{2} \mathbf{u_i}^T \mathbf{k_i}(\tilde{\rho}_i) \mathbf{u_i} = \frac{1}{2} \tilde{\rho}_i{}^p E_0 \mathbf{u_i}^T \mathbf{k_{i,0}} \mathbf{u_i} \tag{1.16}
$$

where $\mathbf{k}_i(\tilde{\rho}_i)$ is computed by scaling the element stiffness matrix by the Young's modulus E_0 (see right side of Eq. 1.16). Meanwhile, the compliance of $i-th$ element can be expressed as:

$$c_i(\tilde{\rho}_i) = \mathbf{u_i}^T \mathbf{k_i}(\tilde{\rho}_i)\mathbf{u_i} = \tilde{\rho}_i^p E_0 \mathbf{u_i}^T \mathbf{k_{i,0}} \mathbf{u_i} = 2 \cdot w_i(\tilde{\rho}_i) \tag{1.17}$$

and the total structural compliance finally is:

$$C(\tilde{\boldsymbol{\rho}}) = \sum_{i=1}^{e_{tot}} c_i(\tilde{\rho}_i) = 2 \sum_{i=1}^{e_{tot}} w_i(\tilde{\rho}_i) \tag{1.18}$$

It is important to note that in Eq. 1.18, e_{tot} includes both design and non-design domain elements, all of which contribute to the total structural compliance. Moreover, as shown in Eq. 1.17, minimizing structural compliance is equivalent to minimizing the total strain energy deformation, which, in other words, corresponds to maximizing structural stiffness.

Compliance gradient

In a stiffness maximization problem (see Eqs. 1.15), the objective function sensitivity can be expressed using the chain rule:

$$\frac{\partial C(\tilde{\boldsymbol{\rho}})}{\partial \rho_i} = \sum_{j \in \Omega^i_{neighbor}} \frac{\partial C(\tilde{\boldsymbol{\rho}})}{\partial \tilde{\rho}_j} \cdot \frac{\partial \tilde{\rho}_j}{\partial \rho_i} \tag{1.19}$$

The derivative of the total compliance w.r.t. filtered element density, using Eq. 1.17, is as following:

$$\frac{\partial C(\tilde{\boldsymbol{\rho}})}{\partial \tilde{\rho}_j} = -p \tilde{\rho}_j^{p-1} E_0 \mathbf{u_j}^T \mathbf{k_{j,0}} \mathbf{u_j} \tag{1.20}$$

In practice, when TO is coupled with commercial FE solvers, the available output is typically the element strain energy $w_i(\tilde{\rho}_i)$ (see Eq. 1.16), which is already computed using the penalized stiffness matrix $\mathbf{k_i}(\tilde{\rho}_i) = \tilde{\rho}_i^p E_0 \mathbf{k_{i,0}}$. Since only the penalized energy might be accessible, the derivative in Eq. 1.20 can be reformulated by eliminating $\mathbf{u_j}^T \mathbf{k_{j,0}} \mathbf{u_j}$ from Eq. 1.16, yielding:

$$\frac{\partial C(\tilde{\boldsymbol{\rho}})}{\partial \tilde{\rho}_j} = -\,p\,\tilde{\rho}_j^{p-1} E_0\, \mathbf{u_j}^T \mathbf{k_{j,0}} \mathbf{u_j} = -\,\frac{2p}{\tilde{\rho}_j}\, w_j(\tilde{\rho}_j). \tag{1.21}$$

The right side of Eq. 1.21 is directly employed to evaluate the compliance derivative w.r.t. $\tilde{\rho}_j$.

Eq. 1.20 (or Eq. 1.21) can be expressed in vectorized form as:

$$\nabla_{\boldsymbol{\rho}} C = \mathbf{H}\, \nabla_{\boldsymbol{\rho}} \mathbf{C} \tag{1.22}$$

where $\nabla_{\boldsymbol{\rho}} \mathbf{C}$ denotes the vector of elemental compliance derivatives, computed according to Eq. 1.20.

Volume constraint gradient

The gradient of the volume constraint, G_v, is given by:

$$\frac{\partial G_v(\tilde{\boldsymbol{\rho}})}{\partial \rho_i} = \sum_{j \in \Omega^i_{neighbor}} \frac{\partial v}{\partial \tilde{\rho}_j} \cdot \frac{\partial \tilde{\rho}_j}{\partial \rho_i} = \sum_{j \in \Omega^i_{neighbor}} v_j \cdot H_{ij} \tag{1.23}$$

where H_{ij} is the filtering weight and v_j is the elemental volume contribution. This gradient remains constant throughout the optimization process, as it depends only on the filtering scheme and the fixed design domain discretization.

Eq. 1.23 can be expressed in vectorized form as:

$$\nabla_{\boldsymbol{\rho}} G_v = \mathbf{H}\, \mathbf{v} \tag{1.24}$$

where $\mathbf{v}$ is the vector collecting the elemental volumes of the design domain.

Filter gradient

The derivative of the filtered density w.r.t. ρ_i is given in a straightforward form as:

$$\frac{\partial \tilde{\rho}_j}{\partial \rho_i} = H_{ij}, \tag{1.25}$$

Eq. 1.25 shows that the filter derivative is exactly given by the filter matrix $\mathbf{H}$.

1.4.2 The stiffness maximization Problem with volume and stress constraints

In the traditional TO formulation, stiffness is maximized for a prescribed amount of material. Traditionally optimized designs often contain high stress concentrations, therefore major manual adjustments or shape optimization is needed to fulfill engineering requirements such as stress constraints.

From the mathematical point of view, the minimum compliance problem with volume

and stress constraints can be formulated as following:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & C(\tilde{\boldsymbol{\rho}}) = \mathbf{U}^T\mathbf{K}(\tilde{\boldsymbol{\rho}})\mathbf{U} = \sum_{i=1}^{e_{tot}} \tilde{\rho}_i^p E_0 \mathbf{u_i}^T \mathbf{k_{i,0}} \mathbf{u_i} \\ \text{s.t.} \quad & \mathbf{K}(\tilde{\boldsymbol{\rho}})\mathbf{U} = \mathbf{F} \\ & G_v = \frac{\sum_{i=1}^{e_{dd}} \tilde{\rho}_i \cdot v_i}{f \cdot V_0} - 1 \leq 0 \\ & G_\sigma = \frac{\sigma_i}{\sigma_{max}} - 1 \leq 0 \\ & \rho_{min} \leq \rho_i \leq 1, \quad \forall i \in \Omega_{\text{dd}} \end{aligned} \tag{1.26}$$

where G_σ denotes the local stress constraint, σ_{max} is the maximum allowable stress, and σ_i is the evaluated stress at the $i-th$ node of the FE model.

Just as in the classical stiffness maximization problem, introducing a stress constraint in TO requires incorporating a stress penalization function. This function penalizes elements with intermediate densities to discourage non-physical stress distributions.

At a generic point a (typically a Gauss integration point), the stress vector contains six components, including both normal and shear stresses:

$$\hat{\sigma}_a(\rho_i) = \begin{pmatrix} \hat{\sigma}_{a,x} \\ \hat{\sigma}_{a,y} \\ \hat{\sigma}_{a,z} \\ \hat{\tau}_{a,xy} \\ \hat{\tau}_{a,yz} \\ \hat{\tau}_{a,zx} \end{pmatrix} \tag{1.27}$$

To account for material interpolation, a penalization function $\eta(\tilde{\rho}(\rho))$ is applied, resulting in the penalized stress:

$$\sigma_a(\rho_i) = \eta(\tilde{\rho}(\rho_i)) \cdot \hat{\sigma}_a(\rho_i) = \tilde{\rho}^\beta \cdot \hat{\sigma}_a(\rho_i) \tag{1.28}$$

where $\beta = 1/2$ [6]. In FEA, the stress at a Gauss point a is typically computed from the constitutive relation:

$$\hat{\sigma}_a(\rho_i) = \mathbf{D}\ \mathbf{B}_a\ \mathbf{u} \tag{1.29}$$

Here:

- $\mathbf{D}$ is the constitutive matrix (e.g., elasticity tensor);
- $\mathbf{B}_a$ is the strain-displacement matrix evaluated at Gauss point a;

- **u** is the nodal displacement vector of the corresponding element.

The penalization ensures that elements with intermediate densities do not artificially carry stress, encouraging binary designs.

Several stress measures can be used in structural analysis; however, the von Mises stress is the most commonly adopted in TO due to its effectiveness in representing yield criteria for ductile materials. The penalized von Mises stress at a stress evaluation point a, denoted as $\sigma_a^{vM}(\boldsymbol{\rho})$, is computed from the corresponding penalized stress tensor introduced earlier, and is given by:

$$\sigma_a^{vM}(\boldsymbol{\rho}) = \sqrt{(\sigma_{a,x}^2 + \sigma_{a,y}^2 + \sigma_{a,z}^2 - \sigma_{a,x}\sigma_{a,y} - \sigma_{a,y}\sigma_{a,z} - \sigma_{a,z}\sigma_{a,x} + 3\tau_{a,xy}^2 + 3\tau_{a,yz}^2 + 3\tau_{a,zx}^2} \tag{1.30}$$

Three different stress constraint approaches are reported in literature [6]:

- **Local:** a stress constraint is applied to each design variable individually. While this provides high accuracy in controlling local stresses, it results in a very high computational cost, making it impractical for large-scale problems;
- **Global:** a single aggregated stress constraint is applied to the entire structure. This approach significantly reduces computational cost, but may be too coarse, potentially overlooking local stress concentrations;
- **Clustered:** the design domain is divided into groups (clusters) of stress evaluation points, and a stress constraint is applied to each cluster. This method offers a balance between accuracy and computational efficiency, allowing for better stress control than the global approach while being less expensive than the local one.

Among these three strategies, the clustered and global approaches are generally preferred in practice. Notably, the global formulation can be seen as a special case of the clustered approach with a single cluster encompassing the entire domain. Since greater emphasis has been placed on the clustered approach, the following formulation will focus on this method.

The primary motivation for using a clustered approach is to significantly reduce the number of stress constraints compared to the local formulation, while still retaining control over local stress behavior. In this method, multiple stress evaluation points are grouped into clusters, and a representative stress measure is computed for each cluster. This is typically done using a modified *p-norm* function, which provides a smooth approximation of the maximum stress within the cluster.

The *p-norm* stress measure for $i-th$ cluster Ω_i, σ_i^{PN}, is defined as [6]:

$$\sigma_i^{PN}(\boldsymbol{\rho}) = \left(\frac{1}{N_i} \sum_{a \in \Omega_i} (\sigma_a^{vM}(\boldsymbol{\rho}))^q \right)^{\frac{1}{q}} \tag{1.31}$$

where N_i is the number of stress evaluation point within Ω_i, σ_a^{vM} is the local von Mises stress (evaluated at Gauss point), and q represents the *p-norm* exponent. This approximation yields a smooth and differentiable function that replaces the behavior of the non-differentiable maximum function. When all local stress values within a cluster are equal, the *p-norm* exactly matches those values. However, in all other cases, it underestimates the true maximum stress.

Increasing the exponent q in the *p-norm* formulation brings the aggregated stress value closer to the maximum local stress within each cluster. However, using excessively high values of q can lead to numerical instabilities and convergence issues. On the other hand, setting $q = 1$ yields the average stress within the cluster. A commonly adopted compromise found in the literature is $q = 8$, which balances accuracy and numerical stability.

Numerical accuracy issues can arise depending on the values of the stress constraint and the exponent p, as the term $(\sigma_a^{vM}(\boldsymbol{\rho}))^q$ may become excessively large. A common remedy is to normalize $\sigma_a^{vM}(\boldsymbol{\rho})$ by the allowable stress σ_{max}, which leads to a mathematically equivalent but more numerically stable formulation. As a result, the stress constraint in the optimization problem is reformulated as:

$$G_\sigma = \left(\frac{\sigma_a^{vM}(\boldsymbol{\rho})}{\sigma_{max}} \right)^p - 1 \leq 0 \tag{1.32}$$

The way clusters are formed, specifically, which evaluation points are included in the set Ω_i, affects the outcome. Two approaches are commonly adopted when forming stress clusters:

- **Stress level approach:** in this method, stress evaluation points with similar stress magnitudes are grouped into the same cluster. This leads to a large variation among the values of the clustered stress measures $\sigma_i^{\mathrm{PN}}(\boldsymbol{\rho})$, but ensures that stresses within each cluster are nearly uniform. As a result, the *p-norm* provides an accurate approximation of the maximum local stress within each cluster. The procedure consists of sorting the stress values at all evaluation points in descending order and then assigning consecutive points to clusters;
- **Distributed Stress Approach:** in this approach, each cluster contains stress evaluation points covering a broad range of stress magnitudes. Consequently, all

clusters tend to have similar average stress levels. The main motivation for this strategy is that it can enhance convergence behavior. High local stresses are effectively dampened by the presence of lower stresses within the same cluster, leading to smoother optimization updates.

Both approaches require the choice of the number of clusters, which is unfortunately problem-dependent and often requires some trial and error. Moreover, even when a stress constraint is explicitly imposed, the optimized solution does not necessarily guarantee that the maximum stress will remain below the prescribed limit.

Stress Gradient

The derivative of the *p-norm* approximation w.r.t. the design variable ρ_i can be computed using the chain rule. It involves the multiplication of three intermediate derivatives:

$$\frac{\partial \sigma_i^{PN}(\boldsymbol{\rho})}{\partial \rho_i} = \sum_{a \in \Omega_i} \frac{\partial \sigma_i^{PN}(\boldsymbol{\rho})}{\partial \sigma_a^{\mathrm{vM}}(\boldsymbol{\rho})} \left(\frac{\partial \sigma_a^{\mathrm{vM}}(\boldsymbol{\rho})}{\partial \boldsymbol{\sigma}_a} \right)^T \frac{\partial \boldsymbol{\sigma}_a(\boldsymbol{\rho})}{\partial \rho_i} \tag{1.33}$$

The gradient is composed of the following three terms:

1. The derivative of the p-norm stress measure w.r.t. the von Mises stress at each evaluation point;
2. The derivative of the von Mises stress w.r.t. the stress components;
3. The derivative of the stress tensor w.r.t. the design variable ρ_i.

The first term:

$$\frac{\partial \sigma_i^{PN}(\boldsymbol{\rho})}{\partial \sigma_a^{vM}(\boldsymbol{\rho})} = \left(\frac{1}{N_i} \sum_{a \in \Omega_i} (\sigma_a^{vM}(\rho))^q \right)^{\frac{1}{q}-1} \frac{1}{N_i} (\sigma_a^{vM}(\rho))^{q-1} \tag{1.34}$$

The second term:

$$\frac{\partial \sigma_a^{vM}(\boldsymbol{\rho})}{\partial \boldsymbol{\sigma}_a} = (\boldsymbol{\sigma}_a(\rho)\mathbf{M}\boldsymbol{\sigma}_a(\rho))^{-1/2}\boldsymbol{\sigma}_a(\rho)\mathbf{M} \tag{1.35}$$

where $\mathbf{M}$ is von Mises stress matrix, defined as:

$$\mathbf{M} = \begin{bmatrix} 1 & -0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 1 & -0.5 & 0 & 0 & 0 \\ -0.5 & -0.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix} \tag{1.36}$$

The last term is not so straightforward like the previous, since it requires the solution of an adjoint model. Expanding the last term in Eq. 1.33, we get:

$$\frac{\partial \boldsymbol{\sigma}_a(\rho)}{\partial \rho_i} = \beta \cdot \tilde{\rho}^{\beta-1} \frac{\partial \tilde{\rho}_j}{\partial \rho_i} \mathbf{D}\mathbf{B}_a\mathbf{u} + \tilde{\rho_i}^{\beta} \mathbf{D}\mathbf{B}_a\mathbf{u} \frac{\partial \mathbf{u}(\rho)}{\partial \rho_i} \tag{1.37}$$

The derivative of displacement $\mathbf{u}(\rho)$ w.r.t. element density can be obtained from the equilibrium equation by differentiating it w.r.t. the well-known element density:

$$\mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \mathbf{F}(\boldsymbol{\rho}) \rightarrow \frac{\partial \mathbf{u}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = -\mathbf{K}^{-1}(\boldsymbol{\rho}) \frac{\partial \mathbf{K}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \mathbf{u}(\boldsymbol{\rho}) \tag{1.38}$$

In Eq. 1.38, is assumed that the derivative of $\mathbf{F}(\boldsymbol{\rho})$ w.r.t. $\boldsymbol{\rho}$ is zero; this assumption is valid only in static load driven problems, where loads are independent from material distribution.

By replacing 1.38 in 1.37, we have the following expression:

$$\frac{\partial \boldsymbol{\sigma}_a(\rho)}{\partial \rho_i} = \beta \cdot \tilde{\rho}^{\beta-1} \frac{\partial \tilde{\rho}_j}{\partial \rho_i} \mathbf{D}\mathbf{B}_a\mathbf{u} - \tilde{\rho_i}^{\beta} \mathbf{D}\mathbf{B}_a\mathbf{K}^{-1}(\boldsymbol{\rho}) \frac{\partial \mathbf{K}(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \mathbf{u}(\boldsymbol{\rho}) \tag{1.39}$$

Substituting 1.34, 1.35 and 1.39 into 1.33, and considering one cluster that contains all the stress evaluation points [18], the *p-norm* gradient becomes:

$$\frac{\partial \sigma^{PN}(\rho)}{\partial \rho} = \frac{\partial \sigma^{PN}(\rho)}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\rho)}{\partial \boldsymbol{\sigma}_a} \left(\beta \cdot \tilde{\rho}^{\beta-1} \frac{\partial \tilde{\rho}_j}{\partial \rho_i} \mathbf{D}\mathbf{B}_a\mathbf{u}(\rho) - \tilde{\rho}^{\beta} \mathbf{D}\mathbf{B}_a\mathbf{K}^{-1}(\rho) \frac{\partial \mathbf{K}(\rho)}{\partial \rho} \mathbf{u}(\rho) \right) \tag{1.40}$$

The gradient of the *p-norm* can be further decomposed for clarity. Specifically, it can be separated into two distinct components based on their dependence: one associated with the stress penalization factor (see Eq. 1.41), and the other with the nodal displacements (see Eq. 1.42).

$$\frac{\partial \sigma_{spf}^{PN}(\rho)}{\partial \rho} = \frac{\partial \sigma^{PN}(\rho)}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\rho)}{\partial \boldsymbol{\sigma}_a} \left(\beta \cdot \tilde{\rho}^{\beta-1} \mathbf{D}\mathbf{B}_a\mathbf{u}(\rho) \right) \tag{1.41}$$

$$\frac{\partial \sigma_u^{PN}(\rho)}{\partial \rho} = \frac{\partial \sigma^{PN}(\rho)}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\rho)}{\partial \boldsymbol{\sigma}_a} \left(-\tilde{\rho}^{\beta} \mathbf{D} \mathbf{B}_a \mathbf{K}^{-1}(\rho) \frac{\partial \mathbf{K}(\rho)}{\partial \rho} \mathbf{u}(\rho) \right) \tag{1.42}$$

By summing 1.41 and 1.42, we get:

$$\frac{\partial \sigma^{PN}(\rho)}{\partial \rho} = \frac{\partial \sigma_{spf}^{PN}(\rho)}{\partial \rho} + \frac{\partial \sigma_u^{PN}(\rho)}{\partial \rho} \tag{1.43}$$

Eq. 1.41 depends on the stress penalization factor and is relatively simple to compute, as it only requires the stress vector and the element density. In contrast, Eq. 1.42 depends on the nodal displacements and is more complex to evaluate. This is due to the need for an explicit definition of the strain-displacement matrix and the global stiffness matrix, both of which are influenced by the specific element formulation.

The solution of 1.42 depends on derivative of displacement w.r.t the element density vector. Since the number of variables in TO is usually large, and adjoint model is required:

$$\mathbf{K}(\boldsymbol{\rho})\boldsymbol{\lambda} = \frac{\partial \sigma^{PN}(\boldsymbol{\rho})}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\boldsymbol{\rho})}{\partial \boldsymbol{\sigma_a}(\boldsymbol{\rho})} \mathbf{D} \mathbf{B}_a \rightarrow \boldsymbol{\lambda} = \frac{\partial \sigma^{PN}(\boldsymbol{\rho})}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\boldsymbol{\rho})}{\partial \boldsymbol{\sigma_a}(\boldsymbol{\rho})} \mathbf{D} \mathbf{B}_a \mathbf{K}^{-}1(\boldsymbol{\rho}) \tag{1.44}$$

where $\boldsymbol{\lambda}$ represents the adjoint variable and it is equal to nodal displacements when the adjoint loads $F_{adj} = \frac{\partial \sigma^{PN}(\boldsymbol{\rho})}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\boldsymbol{\rho})}{\partial \boldsymbol{\sigma_a}(\boldsymbol{\rho})}$ are applied. This nodal displacements can be easily extracted from a FEA by applying the adjoint nodal loads F_{adj}. and solving the equilibrium equation.

Replacing the adjoint variable in 1.42, then leads to:

$$\frac{\partial \sigma_u^{PN}(\rho)}{\partial \rho} = \frac{\partial \sigma^{PN}(\rho)}{\partial \sigma_a^{vM}} \frac{\partial \sigma_a^{vM}(\rho)}{\partial \boldsymbol{\sigma}_a} \left(-\tilde{\rho}^{\beta} \boldsymbol{\xi}_{adj}^{T}(\rho) \frac{\partial \mathbf{K}(\rho)}{\partial \boldsymbol{\rho}} \boldsymbol{\xi}_s(\rho) \right) \tag{1.45}$$

where:

- $\boldsymbol{\xi}_{adj} = \mathbf{B}_a \boldsymbol{\lambda}$ is the deformation vector of the adjoint model;
- $\boldsymbol{\xi}_s = \mathbf{B}_a \mathbf{U}$ is the deformation vector of the state model;
- $\frac{\partial \mathbf{K}(\rho)}{\partial \boldsymbol{\rho}}$ is the derivative of the element stiffness matrix w.r.t. element density.

On the Use of Stress Constraints

While incorporating stress constraints is essential to ensure that the optimized design meets structural requirements, it can also lead to an overly constrained optimization

problem. In such cases, the feasible design space becomes significantly restricted, which may result in difficulties in achieving convergence or even prevent the solver from finding a viable solution. In this work, stress constraints were not included in the final formulation. Despite several attempts, it was not possible to reproduce results consistent with those from ANSYS, primarily due to the inability to exactly match its internal strain-displacement matrix formulation. Multiple efforts were made based on standard isoparametric element formulations from the literature, but discrepancies remained (see appendix .5). As a result, stress-based comparisons were not pursued in this work.

It is also worth noting that even when stress constraints are included in the optimization formulation, there is no guarantee that the final optimized structure will strictly satisfy them. This limitation arises because the local stress constraints are typically represented using an aggregation function (such as the $p - norm$ function), which approximates the maximum stress over the domain. Consequently, some localized stress violations may still occur.

1.4.3 Manufacturing Constraints in Topology Optimization

In practical applications of TO, the resulting design must not only satisfy structural performance requirements but also be manufacturable using the intended production method. Manufacturing constraints are therefore essential to ensure that the optimized geometry is feasible to produce without extensive manual intervention or reinterpretation.

Typical manufacturing constraints include: minimum and maximum feature sizes, symmetry, pattern repetition, extrusion, turning, casting, forging and rolling. The integration of such constraints has become a key area of research [19, 20], especially for large-scale or industrial applications. Without them, the highly complex geometries generated by TO may be impractical to fabricate or convert into clear CAD models.

In this work, particular attention is given to two widely used manufacturing constraints: extrusion and maximum size constraints.

Extrusion constraint

Among various manufacturing techniques, extrusion is widely utilized due to its cost efficiency and its capability to produce components with a constant cross-section. Furthermore, this makes it particularly essential for applications such as ribbed slabs and shell structures in industries including construction, automotive, aerospace, and nuclear fusion. Ribbed shells are widely used in engineering as they enhance structural performance by reducing overall shell thickness. However, designing ribbed shells using conventional engineering methods often results in limited structural configurations, suboptimal

performance, and high material waste. Existing methods that use extrusion manufacturing constraint employ techniques such as anisotropic Helmholtz filters [21] and mapping constraints [22]. These approaches ensure uniformity along the extrusion directions, but often require complex mathematical models or lack practical implementations. The mapping constraint approach, for example, has proven effective in generating ribbed slab designs while relying on a less complex mathematical model. However, it requires extensive preprocessing to group elements that share the same extrusion direction, particularly in curved shell structures. In contrast, the use of Helmholtz filters offers a more general approach, applicable to both structured and unstructured meshes. However, its practical implementation remains challenging, and a vector of extrusion directions must be computed.

Differently from the approach in [21, 22], we developed extrusion constraint based on the element connectivity [23]. Its mathematical formulation, as well as its implementation, are discussed below.

The extrusion constraint ensures uniform material distribution along a predefined direction (or directions in the case of a curved body) in a structured mesh. This constraint is essential for manufacturability in extrusion-based processes and is also particularly useful for designing ribbed layouts; The proposed method leverages the connectivity of the structured mesh to identify and enforce the extrusion direction using a systematic approach.

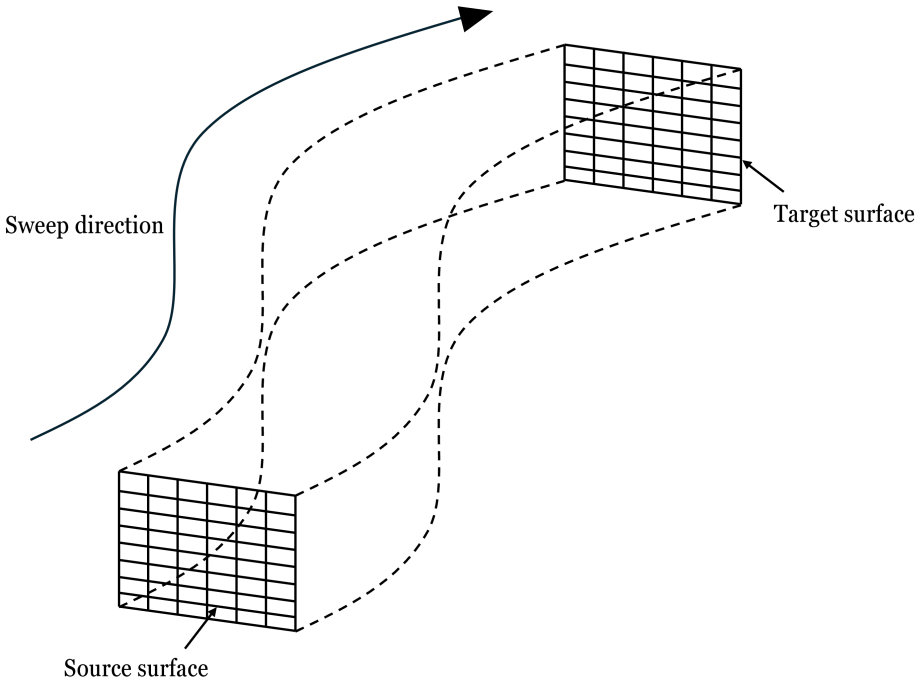


Figure 1.4: Example of structured mesh obtained using sweep method.

A structured mesh is composed of hexahedral (either a combination of hexahedral and

pyramidal, either only pyramidal) elements generated using the sweep method, where source and target surfaces can be clearly identified (Fig. 1.4). Due to their favorable numerical properties, hexahedral meshes are widely used in numerical simulations across engineering disciplines. Their structured nature offers higher accuracy and convergence rates in FEA than tetrahedral meshes, particularly for bending and contact problems. Additionally, they reduce numerical dispersion and locking effects, improving efficiency in structural, fluid, and multi-physics simulations. Their better aspect ratio control further minimizes the number of elements needed for a given accuracy level [24].

Let Ω be the domain which is composed of design domain (Ω_{dd}) and non-design domain elements (Ω_{ndd}), discretized into n elements. The extrusion constraint requires that, elements along the same extrusion direction maintain consistent material densities. This can be mathematically expressed as:

$$\tilde{\rho}_i = \tilde{\rho}_j, \quad \forall i, j \in \mathcal{C}_k, \quad k = 1, 2, \ldots, l \tag{1.46}$$

where:

- $\tilde{\rho}_i$: $i - th$ filtered element density associated with those belonging on the base surface;
- $\tilde{\rho}_j$: $j - th$ filtered element density associated with those sharing the same extrusion column as $\tilde{\rho}_i$;
- $\mathcal{C}_k$: set of elements aligned along the $k - th$ extrusion column;
- l: total number of extrusion columns.

The mapping constraint in Eq. 1.46 enforces the elements belonging in the same extrusion direction to have the same density as the corresponding element on the base surface. The base surface, also referred as reference surface, divides the non-design and design domains, and its identification is crucial for applying this method. Once the base surface is defined, the design elements belonging on it become automatically the base elements used to impose the extrusion constraint, so equal element densities within each $\mathcal{C}_k$.

To enforce the extrusion constraint, we construct a binary matrix $\mathbf{H}_{\text{extrusion}} \in \mathbb{R}^{e_{dd} \times e_{dd}}$ that represents the extrusion connectivity:

$$H^{ij}_{extrusion} = \begin{cases} 1, & \text{if elements } i \text{ and } j \text{ are in the same extrusion column,} \\ 0, & \text{otherwise.} \end{cases} \tag{1.47}$$

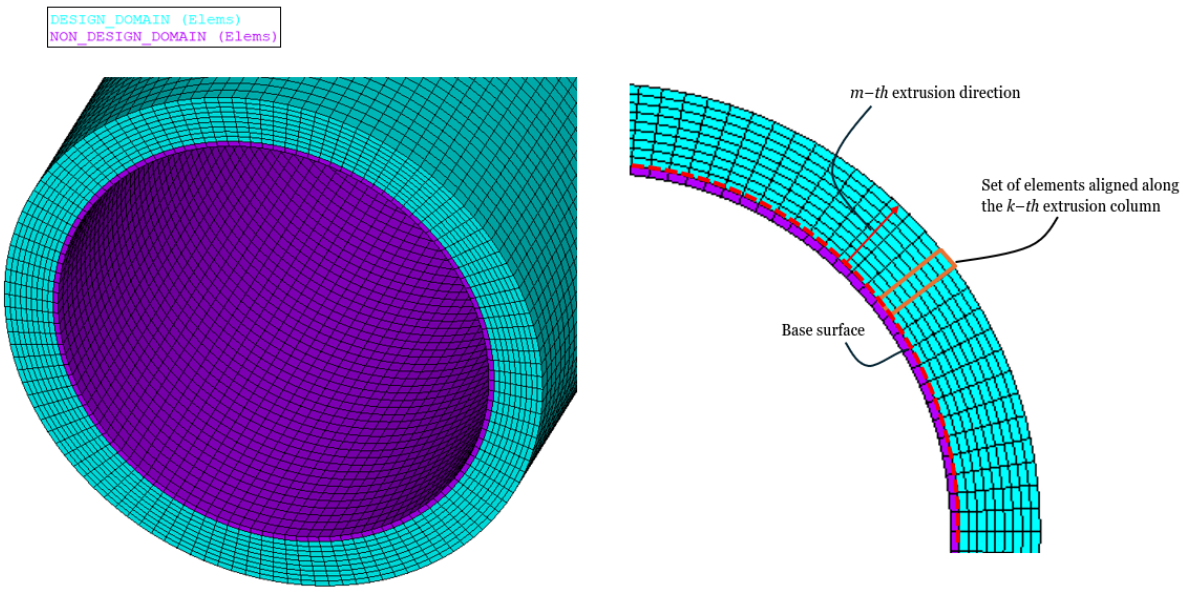


Figure 1.5: Example of structured mesh with the base surface (red dot line), $k-th$ extrusion direction and set of elements sharing common extrusion direction ($\mathcal{C}_k$).

The $\mathbf{H}_{\text{extrusion}}$ matrix is a square matrix and its construction is described by the following pseudo-code:

1. Select a surface as the base surface. This surface divides the design and non-design domains.
2. Identify all extrusion columns $\mathcal{C}_k$ by following adjacency relationships along the extrusion direction.
3. Construct $\mathbf{H}_{\text{extrusion}}$ matrix:
   - 3.1 For each extrusion column $\mathcal{C}_k$:
     - 3.2 For each element $i \in \mathcal{C}_k$:
       - 3.3 Set $H^{ij}_{\text{extrusion}} = 1, \quad \forall j \in \mathcal{C}_k$.

A clear understanding of how the neighbor adjacent elements search is performed for linear hexahedral elements is shown in Fig. 1.6, where a column of design and non-design elements and the base surface are depicted. Starting from element e_2, the shared nodes with its adjacent element are n_9, n_{10}, n_{11} and n_{12}; therefore, element e_3 will have the same extrusion direction as e_2. This process is repeated for element e_3 and continues until no more common shared nodes exist.

Now, let $\tilde{\boldsymbol{\alpha}}$ represent the filtered sensitivity vector of the objective function with respect to the filtered element density vector $\tilde{\boldsymbol{\rho}}$. The average sensitivity, as suggested in [22], can be computed as follows:

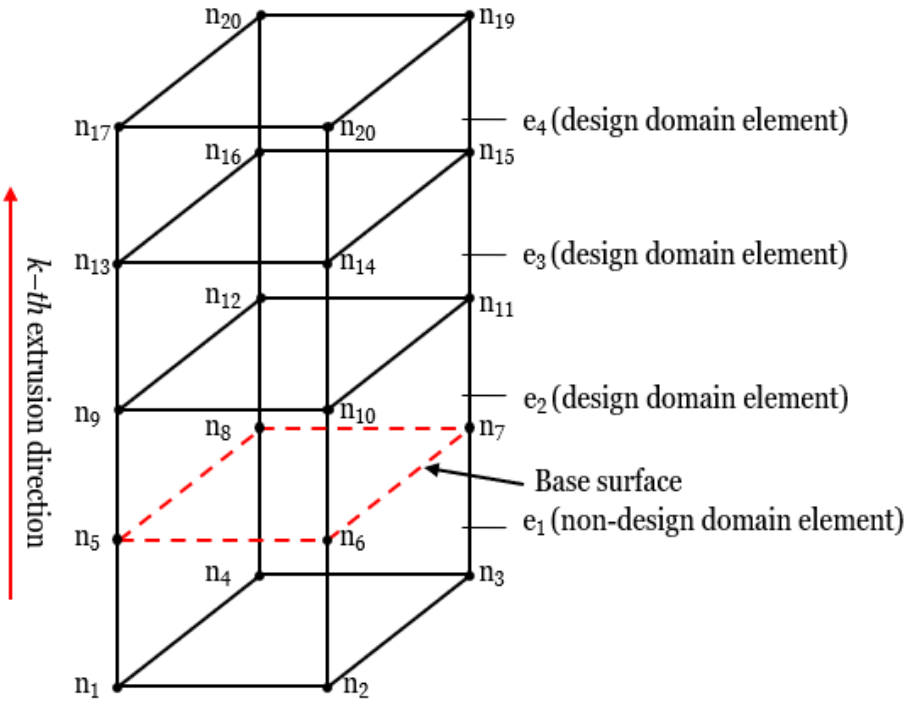


Figure 1.6: Example of column elements that share the same extrusion direction.

$$\boldsymbol{\alpha_{avg}} = \frac{\mathbf{H_{extrusion}}\ \tilde{\boldsymbol{\alpha}}}{n_{layer}} \tag{1.48}$$

where n_{layer} in Eq. 1.48 represents the number of elements along $k-th$ extrusion direction. In other words, it is the number of elements within the $\mathcal{C}_k$ set. Since the mesh is obtained with the sweep method, the number of elements will be constant for each $\mathcal{C}_k$ set.

Multiple Extrusion constraint

The multiple extrusion constraint introduced in this work represents a natural extension of the single extrusion constraint described in sec. 1.4.3. The principle remains the same: a uniform material distribution is enforced along an extrusion direction, but now this is applied independently within different subdomains of the FE model. For each subdomain, a reference (or base) surface is first identified, and the extrusion matrix $\mathbf{H}_{\text{extrusion}}$ is then constructed according to the pseudo-code provided in sec. 1.4.3, with the additional rule that elements outside the current subdomain are set to zero in $\mathbf{H}_{\text{extrusion}}$. This strategy is particularly useful because, if only a single extrusion constraint were applied to the entire model, the optimization would tend to produce an "averaged" structural response under the applied loads. By applying multiple extrusion constraints, the optimization is able to capture localized effects more accurately and identify additional structural features that contribute to a more effective and realistic design. This methodology were applied

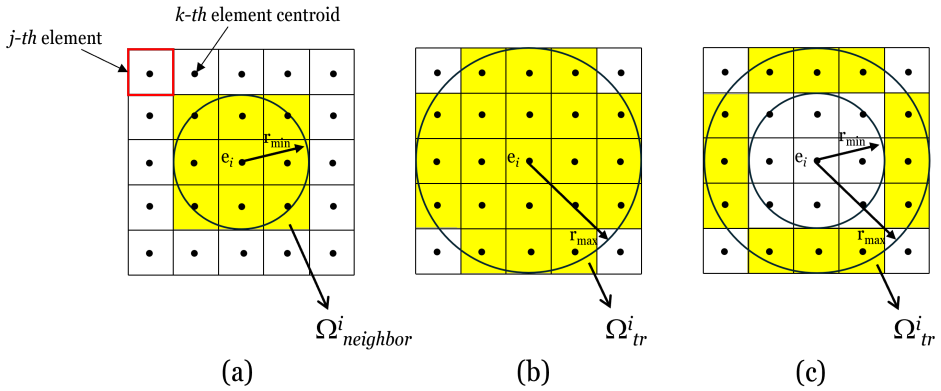


Figure 1.7: Sketch of a 2D filter used to impose: (a) minimum length scale and neighbor filtering; (b) maximum size constraint [2]; (c) maximum size constraint [3].

in chapter 4, to achieve a conceptual design of magnet's reinforcements.

Maximum Size constraint

In TO, simplistic loading conditions or excessive material usage often result in the concentration of material in localized regions. This leads to unclear or poorly defined reinforcement structures. To address this issue and achieve well-defined reinforcement patterns, the maximum size constraint is used to control the spatial distribution of material by limiting the maximum size of material features in the design domain. The constraint formulation was first proposed by Guest [2], and it is still the reference for most TO codes. The maximum size constraint is typically enforced through the concept of local test regions resulting in several local constraints within Ω_{dd} domain (Fig. 1.7b). In 3D problems, a spherical volume is used as the test region, defined by an outer radius $d_{max}/2$ equal to the maximum member size desired. Fernàndez et al. [3] proposed also the use of a spherical shell as the test region (Fig. 1.7c)), defined by an inner radius $d_{min}/2$ and an outer radius $d_{max}/2$. This approach, which is also implemented in this work, minimizes the introduction of holes in the topology, thereby enhancing the manufacturability. For each element in Ω_{dd} the associated test region Ω^i_{tr} is defined as:

$$\Omega^i_{tr} = \left\{ e_j \;\middle|\; \frac{d_{\min}}{2} < \|e_j - e_i\| \leq \frac{d_{\max}}{2}, \quad \forall i, j \in \Omega_{\text{dd}} \right\} \tag{1.49}$$

For each test region Ω^i_{tr}, the local maximum size constraint is defined as:

$$g_i = \psi - \frac{\sum\limits_{i \in \Omega_{tr}} v_i (1 - \tilde{\rho}_i)^p}{\sum\limits_{i \in \Omega_{tr}} v_i} \leq 0, \quad \forall i \in \Omega_{\text{dd}} \tag{1.50}$$

where ψ is a small positive value that represents the minimum allowable void ratio ($\psi = 0.05$). This void ratio acts as a buffer, preventing structural members from achieving diameter d_{max}. The larger ψ, the smaller will be the structural members within Ω_{dd}. The exponent p determines the extent to which elements with intermediate volume fractions contribute to satisfying the volume of voids requirement, and it is usually set as in the SIMP method [2]. Eq. 1.50 can be formulated in vectorized form, which is easier for implementation purposes:

$$\mathbf{g} = \psi - \mathbf{H_{tr}}\boldsymbol{\delta} \tag{1.51}$$

where $\mathbf{H_{tr}}$ is the test region matrix, while the vector $\boldsymbol{\delta}$ groups the measure of void in each test region Ω^i_{tr}. $\mathbf{H_{tr}}$ components are computed as:

$$H^{ij}_{tr} = \frac{v_i}{\sum\limits_{i \in \Omega_{tr}} v_i}, \quad \forall i \in \Omega_{\text{dd}} \tag{1.52}$$

meanwhile, the components of $\boldsymbol{\delta}$ are defined as:

$$\delta_i = (1 - \tilde{\rho}_i)^p \tag{1.53}$$

Due to the large number of constraints, the local maximum size constraints are aggregated using *p-mean* or *p-norm* function P, which smooths the individual constraints:

$$P = \begin{cases} P_{\text{mean}} = \left(\frac{1}{e_{dd}} \sum\limits_{i \in \Omega_{\text{dd}}} (s_i)^{p_{agg}} \right)^{\frac{1}{p_{agg}}}, & if \quad P = P_{\text{mean}} \\ P_{\text{norm}} = \left(\sum\limits_{i \in \Omega_{\text{dd}}} (s_i)^{p_{agg}} \right)^{\frac{1}{p_{agg}}}, & if \quad P = P_{\text{norm}} \end{cases} \tag{1.54}$$

where e_{dd} is the number of elements in Ω_{dd}, p_{agg} is the aggregation exponent that governs the accuracy of the aggregation, s_i denotes the *Quantity of Interest* being aggregated. p_{agg} is a large number (greater than one hundred) to capture the most violated maximum size constraint; in fact, when a large exponent is used and s_i takes positive values, the aggregation function closely approximates the maximum value. Additionally, the key distinction between these functions lies in their estimation behavior: the *p-norm* tends to overestimate the maximum value, whereas the *p-mean* tends to underestimate it. The primary advantage of these aggregation functions is their ability to provide sensitivity information, which is crucial for gradient-based optimization methods. In this work, we adopt the *p-norm* aggregation function instead of the *p-mean*, as it provides a more

accurate approximation of the most violated constraint when a large exponent is used. Specifically, we set $p_{agg} = 300$ to effectively capture the most critical constraint violations. While both functions have been studied in the literature, a detailed comparison of their properties can be found in [3]. Since the primary focus of this study is on the integration of manufacturing constraints in TO rather than on the choice of aggregation function, we do not present an extensive analysis of their differences here. However, for further extensions of this research, we have considered both formulations and included them in our manuscripts.

In Eq. 1.55 the *Quantity of Interest* is defined as:

$$s_i = g_i + 1 - \psi, \qquad \forall i \in \Omega_{\text{dd}} \tag{1.55}$$

which ranges between 0 and 1. Using P, which can represent either the *p-mean* or *p-norm* function, the global constraint G_{ms} is replaced as follows:

$$G_{ms} = P - 1 + \psi \leq 0 \tag{1.56}$$

In this work, due to the enforcement of the extrusion constraint, which ensures uniform element density within each $\mathcal{C}_k$ set, the test regions are defined exclusively for elements located on the base surface, therefore the local constraints g_i are only evaluated for these elements. This significantly reduces the computational effort.

Maximum Size constraint gradient

Differently from the objective and global volume constraint function gradients, the maximum size constraint gradient requires more terms to be evaluated. Its gradient is computed using the chain rule:

$$\frac{\partial G_{ms}(\tilde{\boldsymbol{\rho}})}{\partial \rho_i} = \frac{\partial \tilde{\rho}_i}{\partial \rho_i} \frac{\partial \delta_i}{\partial \tilde{\rho}_i} \frac{\partial g_i}{\partial \delta_i} \frac{\partial s_i}{\partial g_i} \frac{\partial P}{\partial s_i} \frac{\partial G_{ms}}{\partial P} \tag{1.57}$$

The derivative of the measure of void δ_i with respect to the filtered element density $\tilde{\rho}_i$ is as follows (see Eq. 1.53):

$$\frac{\partial \delta_i}{\partial \tilde{\rho}_i} = -p \cdot (1 - \tilde{\rho}_i)^{p-1} \tag{1.58}$$

The derivative of the local maximum size constraint g_i with respect to $\tilde{\rho}_i$ is given by (see Eq. 1.50):

$$\frac{\partial g_i}{\partial \delta_i} = -\frac{v_i}{\sum\limits_{i \in \Omega_{tr}} v_i} \tag{1.59}$$

The derivative of the *Quantity of Interest*, s_i, with respect to g_i is expressed as follows (see Eq. 1.55):

$$\frac{\partial s_i}{\partial g_i} = 1 \tag{1.60}$$

The derivative of the aggregation function P with respect to s_i is given by (see Eq. 1.54):

$$\frac{\partial P}{\partial s_i} = \begin{cases} \frac{1}{n_{dd}} \cdot s_i^{p_{agg}-1} \cdot P_{mean}^{1-p_{agg}}, & \text{if} \quad P = P_{\text{mean}} \\ s_i^{p_{agg}-1} \cdot P_{mean}^{1-p_{agg}}, & \text{if} \quad P = P_{\text{norm}} \end{cases} \tag{1.61}$$

Finally, the derivative of the aggregated maximum size constraint G_{ms} with respect to P is computed as (see Eq. 1.56):

$$\frac{\partial G_{ms}}{\partial P} = 1 \tag{1.62}$$

About the use of Maximum Size constraint

In sec. 1.4.3 we presented the analytical formulation of the maximum size constraint and its gradient. This manufacturing constrain was initially introduced for allowing smaller geometrical features in large scale magnets, typical of a fusion power plant. However, even if not directly applied in the context of magnet reinforcement design, is intentionally described for a future use where is essential to constraint the size of the components for easier manufacturability, handling and assembly.

1.5 Updating Design Variables: Optimization Algorithms

The optimization problem that comes from TO can be addressed using:

- Optimality Criteria (OC);
- Sequential Quadratic Programming (SQP);
- Method of Moving Asymptotes (MMA).

The nature of the problem's approximation depends on the type of algorithm. Both SQP and MMA are general method that can address several constrained optimization problems, but they are complex to implement, meanwhile OC is simpler, but it works only for few problems (like stiffness maximization under volume constraints) and it suffers from slow convergence requiring more iterations. Therefore, to solve more case of structural optimization problems, it is preferred to use either SQP or MMA, despite their complexity.

Since the MMA algorithm is used in this work, a brief overview of its key principles is provided below.

1.5.1 Method of Moving Asymptotes (MMA)

A general optimization problem is described as following:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq b_i, \quad i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, n \end{aligned} \tag{1.63}$$

The optimization problem 1.63 aims to find the design variable (or set of variables) $\mathbf{x}$ that minimizes a given objective function $f(\mathbf{x})$, subject to a set of m inequality constraints $g_i(\mathbf{x}) \leq b_i$ and n equality constraints $h_j(\mathbf{x}) = 0$. In the context of structural TO, this general formulation is commonly expressed as following:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq b_i, \quad i = 1, \ldots, m \\ & \mathbf{K}(\mathbf{x})\mathbf{U} = \mathbf{F}(\mathbf{x}) \\ & x_j^{min} \leq x_j \leq x_j^{max}, \quad j = 1, \ldots, n \end{aligned} \tag{1.64}$$

where $g_i(\mathbf{x})$ denotes the constraint functions associated with the final design (e.g., limits on material usage or allowable stress), $\mathbf{K}(\mathbf{x})\mathbf{U} = \mathbf{F}(\mathbf{x})$ represents the equilibrium equation solved at each iteration, and $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper bounds of the DVs, respectively.

Structural optimization problems are usually coupled with FEA, with up to millions of degrees of freedom leading to nonconvex form, which makes finding their explicit solution computationally expensive or in most cases either impractical or impossible. To face these limitations, an approach to approximate large scale problems as a sequence of explicit, convex subproblems was proposed by Krister Svanberg in 1987 [25] through the Method of Moving Asymptotes (MMA).

MMA is a gradient-based optimization algorithm that requires the evaluation of gradients for both objective and constraint functions at each iteration. MMA addresses challenging nonlinear and potentially non-convex optimization problems by replacing them with a sequence of convex approximation subproblems, which are significantly easier to solve. The construction of these subproblems is guided by dynamically adjusted parameters known as moving asymptotes, which help control the shape and curvature of the approximations throughout the iterative process.

The iterative scheme of MMA is composed of the following steps:

1. choose a starting point $\mathbf{x}^0$, that represents the vector of DVs;
2. given an iteration point $\mathbf{x}^k$, calculate the constraint functions $f_i(\mathbf{x}^k)$ and their gradients $\nabla f_i(\mathbf{x}^k)$;
3. generate an explicit subproblem P^k which approximates the original problem 1.63;
4. solve P^k and let the optimal solution of this subproblem be the next iteration point $\mathbf{x}^{k+1}$. Then go to step 2 and repeat until a convergence criteria is fulfilled.

The approximating functions replaced in P^k at $\mathbf{x}^k$ design point, are as following:

$$f_i^k = \sum_{j=1}^{n} \left(\frac{p_{ij}^k}{U_j^k - x_j} + \frac{q_{ij}^k}{x_j - L_j^k} \right) + r_i^k, \quad i = 1, \ldots, m \tag{1.65}$$

where L_j^k and U_j^k represent the so called lower and upper *moving asymptotes*, which limit each DVs such that:

$$L_j^k \leq x_j^k \leq U_j^k \tag{1.66}$$

while the terms p_{ij}^k, q_{ij}^k and r_i^k are defined as following:

$$p_{ij}^k = \begin{cases} (U_j^k - x_j^k)^2 \frac{\partial f_i}{\partial x_j}, & \text{if} \quad \frac{\partial f_i}{\partial x_j} > 0 \\ 0, & \text{if} \quad \frac{\partial f_i}{\partial x_j} \leq 0 \end{cases} \tag{1.67}$$

$$q_{ij}^{(k)} = \begin{cases} 0, & \text{if} \quad \frac{\partial f_i}{\partial x_j} \geq 0 \\ -(x_j^k - L_j^k)^2 \frac{\partial f_i}{\partial x_j}, & \text{if} \quad \frac{\partial f_i}{\partial x_j} < 0 \end{cases} \tag{1.68}$$

$$r_i^k = f_i^k - \sum_{j=1}^{n} \left(\frac{p_{ij}^k}{U_j^k - x_j} + \frac{q_{ij}^k}{x_j - L_j^k} \right) \tag{1.69}$$

where the derivatives $\frac{\partial f_i}{\partial x_j}$ are evaluated at design point $\mathbf{x} = \mathbf{x}^k$. The expression of f_i^k is a first order approximation of f_i at $\mathbf{x}^k$. Since $p_{ij}^k \geq 0$ and $q_{ij}^k \geq 0$, the approximating function f_i^k is convex. Considering the design point $\mathbf{x} = \mathbf{x}^k$, the second derivative of f_i^k is:

$$\frac{\partial^2 f_i^k}{\partial x_j^2} = \begin{cases} \frac{2\frac{\partial f_i}{\partial x_j}}{U_j^k - x_j^k}, & \text{if} \quad \frac{\partial f_i}{\partial x_j} > 0 \\ -\frac{2\frac{\partial f_i}{\partial x_j}}{x_j^k - L_j^k}, & \text{if} \quad \frac{\partial f_i}{\partial x_j} \leq 0 \end{cases} \tag{1.70}$$

Therefore, the closer the bounds L_j^k and U_j^k are set to the current design variable x_j^k, the larger the resulting second derivatives become. This increases the curvature of the approximating functions f_i^k, making the approximation of the original problem more conservative.

Dual method to solve the subproblems

The MMA subproblem (P^k) at k-th iteration is now defined as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{j=1}^{n} \left(\frac{p_{0j}}{U_j - x_j} + \frac{q_{0j}}{x_j - L_j} \right) + r_0 \\ \text{s.t.} \quad & \sum_{j=1}^{n} \left(\frac{p_{ij}}{U_j - x_j} + \frac{q_{ij}}{x_j - L_j} \right) \leq b_i, \quad i = 1, \ldots, m \\ & \alpha_j \leq x_j \leq \beta_j, \quad j = 1, \ldots, n \end{aligned} \tag{1.71}$$

where the parameters α_j^k and β_j^k are the so called *move limits* that aim at avoiding division by zero when solving the subproblem. The subproblem 1.71 is a convex and separable problem, meaning that a dual method can be employed for its solution. If the primal problem is a minimization, its corresponding dual will be a maximization problem (and vice versa). According to the principle of weak duality, any feasible solution of the dual provides a lower bound to the objective value of the primal, and any feasible solution of the primal gives an upper bound to the objective value of the dual. In other words, the value of the objective function in the dual problem never exceeds that of the primal when both are evaluated at feasible points.

The Lagrangian function of the subproblem 1.71 is defined as:

$$l(x, y) = f_0^k(\mathbf{x}) + \sum_{i=1}^{m} y_i f_i^k(\mathbf{x}) \tag{1.72}$$

which can be rewritten as:

$$l(x,y) = r_0 + \mathbf{y^T b} + \sum_{j=1}^{n}\left(\frac{p_{0j} + \mathbf{y^T p_j}}{U_j - x_j} + \frac{q_{0j} + \mathbf{y^T q_j}}{x_j - L_j}\right) = r_0 - \mathbf{y^T b} + \sum_{j=1}^{n} \mathbf{l_j}(x_j, \mathbf{y}) \quad (1.73)$$

$\mathbf{y}$ in Eq. 1.73 is the vector of Lagrange multipliers.

Using the weak duality property, a dual objective function $W(\mathbf{y})$ is defined for all $y_i \geq 0$:

$$W(\mathbf{y}) = \min_{\mathbf{x}} \{l(\mathbf{x}, \mathbf{y}) \; ; \; \alpha_j \leq x_j \leq \beta_j \;\text{ for all } j\} = r_0 - \mathbf{y}^\top \mathbf{b} + \sum_{j=1}^{n} W_j(\mathbf{y}) \quad (1.74)$$

where:

$$W_j(\mathbf{y}) = \min_{\mathbf{x_j}} = \{l_j(x_j, \mathbf{y}) \; \alpha_j \leq x_j \leq \beta_j\} \quad (1.75)$$

Since $\mathbf{y} \geq 0$, it follows that $p_{0j} + \mathbf{y^T p}_j \geq 0$ and $q_{0j} + \mathbf{y^T q}_j \geq 0$, therefore $l_j(x_j, \mathbf{y})$ is convex.

The first and second derivatives of $l_j(x_j, \mathbf{y})$ w.r.t. x_j are, respectively:

$$\frac{\partial l_j(x_j, \mathbf{y})}{\partial x_j} = \frac{p_{0j} + \mathbf{y}^\top \mathbf{p}_j}{(U_j - x_j)^2} - \frac{q_{0j} + \mathbf{y}^\top \mathbf{q}_j}{(x_j - L_j)^2} \quad (1.76)$$

$$\frac{\partial^2 l_j(x_j, \mathbf{y})}{\partial x_j^2} = \frac{2(p_{0j} + \mathbf{y}^\top \mathbf{p}_j)}{(U_j - x_j)^3} + \frac{2(q_{0j} + \mathbf{y}^\top \mathbf{q}_j)}{(x_j - L_j)^3} \quad (1.77)$$

Since $\frac{\partial^2 l_j(x_j, \mathbf{y})}{\partial x_j^2}$ is strictly positive, $\frac{\partial l_j(x_j, \mathbf{y})}{\partial x_j}$ is strictly increasing in x_j and the following statements are verified:

1. If $\frac{\partial l_j(\alpha_j, \mathbf{y})}{\partial x_j} \geq 0$ then $x_j(\mathbf{y}) = \alpha_j$ (17)
2. If $\frac{\partial l_j(\beta_j, \mathbf{y})}{\partial x_j} \leq 0$ then $x_j(\mathbf{y}) = \beta_j$ (18)
3. If $\frac{\partial l_j(\alpha_j, \mathbf{y})}{\partial x_j} < 0$ then $x_j(\mathbf{y})$ is the unique solution of the equation $\frac{\partial l_j(x_j, \mathbf{y})}{\partial x_j} = 0$.

It can be verified that the unique solution $x_j(\mathbf{y})$ satisfying $\frac{\partial l_j(x_j,\mathbf{y})}{\partial x_j} = 0$ is given by:

$$x_j(\mathbf{y}) = \frac{\left(p_{0j} + \mathbf{y}^\top \mathbf{p}_j\right)^{1/2} L_j + \left(q_{0j} + \mathbf{y}^\top \mathbf{q}_j\right)^{1/2} U_j}{\left(p_{0j} + \mathbf{y}^\top \mathbf{p}_j\right)^{1/2} + \left(q_{0j} + \mathbf{y}^\top \mathbf{q}_j\right)^{1/2}} \tag{1.78}$$

Given that an explicit expression for $x_j(\mathbf{y})$ exists, the corresponding expression for the dual objective function $W(\mathbf{y})$ can be written as:

$$W(\mathbf{y}) = r_0 - \mathbf{y}^\top \mathbf{b} + \sum_{j=1}^{n} \left(\frac{p_{0j} + \mathbf{y}^\top \mathbf{p}_j}{U_j - x_j(\mathbf{y})} + \frac{q_{0j} + \mathbf{y}^\top \mathbf{q}_j}{x_j(\mathbf{y}) - L_j} \right) \tag{1.79}$$

Furthermore, the partial derivatives of $W(\mathbf{y})$ w.r.t. the dual variables y_i are given by:

$$\frac{\partial W}{\partial y_i} = -b_i + \sum_{j=1}^{n} \left(\frac{p_{ij}}{U_j - x_j(\mathbf{y})} + \frac{q_{ij}}{x_j(\mathbf{y}) - L_j} \right) \tag{1.80}$$

The dual problem associated with subproblem $P^{(k)}$ is to maximize the function $W(\mathbf{y})$ (considering $\mathbf{y} \geq 0$)

$$\text{D:} \quad \max W(\mathbf{y}) \quad \text{s.t.} \quad \mathbf{y} \geq 0 \tag{1.81}$$

Since $x_j(\mathbf{y})$ is a continuous function of $\mathbf{y}$ (except in the degenerate case where $p_{0j} + \mathbf{y}^\top \mathbf{p}_j = 0$ and $q_{0j} + \mathbf{y}^\top \mathbf{q}_j = 0$), the dual objective $W(\mathbf{y})$ is smooth. Moreover, because it represents the point wise minimum of a family of affine functions in $\mathbf{y}$, it is also concave.

Consequently, problem D in Eq. 1.81 is well-behaved and suitable for gradient-based optimization methods. Once the dual problem has been solved, the optimal solution to the primal subproblem $P^{(k)}$ is easily recovered by substituting the optimal dual variables $\mathbf{y}$ into the analytical expression for $x_j(\mathbf{y})$ given in Eq. 1.78.

1.5.2 About normalization in optimization problems

It is important to note that the sensitivities of both objective and constraint functions are unit-dependent. This means that the magnitude of an elemental sensitivity can vary significantly depending on the units in which it is expressed. As a consequence, sensitivities related to one constraint might dominate others, potentially leading to an ill-conditioned and unbalanced optimization process.

To address this, normalization is employed to improve both the numerical stability and performance of the optimization algorithm. In particular, normalization is essential for the following reasons:

- Numerical stability: DVs and sensitivities may differ in scale by several orders of magnitude, which can introduce instability. Normalization brings all quantities to a comparable scale, mitigating this risk;
- Improved convergence: normalized variables contribute to a smoother and more well-behaved optimization landscape, aiding convergence, especially for gradient based algorithms such as MMA or SQP;

Normalization typically involves scaling objective functions, constraints, and their gradients using appropriate reference values—such as their initial magnitudes or known feasible bounds.

In this work, normalization was consistently applied to both objective and constraint functions, as well as their respective gradients. Mathematically, the normalization of the objective function $F(\tilde{\boldsymbol{\rho}})$ is performed w.r.t. its initial value at the first iteration:

$$F_{\text{norm}}(\tilde{\boldsymbol{\rho}}) = \frac{F^{(k)}(\tilde{\boldsymbol{\rho}})}{F^{(k=1)}(\tilde{\boldsymbol{\rho}})} \tag{1.82}$$

where k denotes the current iteration.

For a generic constraint function G, gradient normalization is performed as follows:

$$\frac{\partial G_{norm}}{\partial \tilde{\boldsymbol{\rho}}} = \frac{\partial G}{\partial \tilde{\boldsymbol{\rho}}} \Big/ ||\nabla G|| \tag{1.83}$$

where the gradient norm $||\nabla G||$:

$$||\nabla G|| = \sqrt{\sum_i \left(\frac{\partial G}{\partial \tilde{\rho}_i}\right)^2} \tag{1.84}$$

This normalization strategy ensures that all constraints are appropriately weighted and contribute meaningfully to the optimization process, without being biased by differences in scale or units.

1.6 PyMapdl Topology Optimization Tool

In this work, a TO tool has been implemented into the PyMAPDL class within the ANSYS suite. The package provides a Python-friendly interface to control the software, enabling the execution of low-level APDL commands and facilitating data exchange via high-performance gRPC interfaces [26]. The open access code on how Python and APDL

are coupled for TO applications can be found here [27]. This code provides an intuitive way to implement the TO framework in PyMAPDL. Other interesting open-access code that couples the Abaqus FE software and Python is [28]. Here, the stress constraint is also integrated into the TO formulation.

The developed PyMAPDL TO code is organized into several sections, as illustrated in the flowchart in Fig. 1.8. The main sections are described below.

- **Parameter Settings**: this section defines the primary TO parameters, such as the mechanical properties (Young's modulus of solid (E_0) and Poisson's ratio (ν)), the maximum number of iterations, the convergence tolerance, filter size, and TO constraints.

- **Initialization**: this phase stores information about the mesh, design elements, and non-design elements. These details are subsequently used for calculating the minimum filter radius and constructing the filter (Eq. 1.13), extrusion (pseudo-code in sec. 1.4.3) and test region matrices (Eq. 1.49).

- **Solver**: During this iterative phase, the optimization problem is solved using the MMA algorithm [25], chosen for its versatility, widespread application in TO problems, and availability in Python [29]. The process follows these key steps:

    1. *Initialize MMA parameters* and set the initial element densities to the prescribed volume fraction.
    2. *Material Property Update*: compute the element Young's modulus based on the density values using Eq. 1.1.
    3. *Density Filtering*: apply the density filter through Eq. 1.11 to smooth the element densities.
    4. *Finite Element Analysis (FEA)*: solve the equilibrium equation $\mathbf{K}(\tilde{\boldsymbol{\rho}})\mathbf{U} = \mathbf{F}$ using ANSYS FE software.
    5. *Objective and Constraint Evaluation*: compute the objective function (Eq. 1.18), constraints (Eq. 1.15, Eq. 1.56), and their respective gradients (Eq. 1.23, Eq. 1.57).
    6. *Gradient filtering*: apply the filter to the objective gradient (Eq. 1.19) and compute the average objective function for each $\mathcal{C}_k$ set (Eq. 1.48).
    7. *Optimization Step*: update the element density vector $\boldsymbol{\rho}$ using the MMA optimizer.
    8. *Convergence Check*: if convergence criteria are met (Eq. 1.86), the process stops; otherwise, the loop repeats from step 2 to 8.

- **Postprocessing**: in the final stage, element density plots are generated and saved as ".vtk" format, and maximum stress and displacement values are recorded and printed.

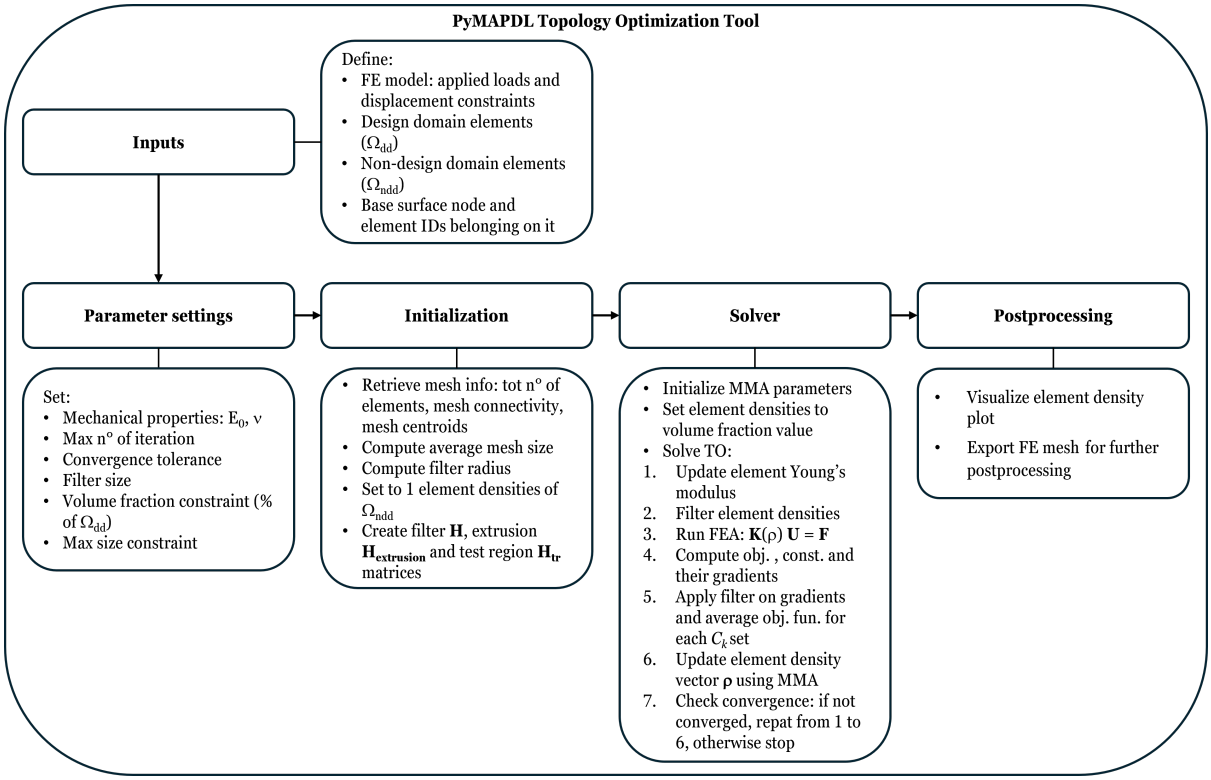


Figure 1.8: Flowchart of PyMAPDL Topology Optimization Tool.

The code has been developed with flexibility in mind, enabling it to be easily updated to accommodate other TO problem formulations. This includes defining alternative objective and constraint functions, manufacturing constraints, as well as implementing different types of filters and material law interpolations, ensuring adaptability to various optimization scenarios.

Since TO is an iterative method, a convergence criterion must be established. In this work, the convergence check is based on the stability of the objective function over recent iterations. Specifically, a moving average of the objective function values from the last five iterations is computed as:

$$\bar{f}\text{obj} = \frac{1}{5} \sum_{i=it-4}^{it} f_{\text{obj}}^{i} \tag{1.85}$$

The absolute difference between the current objective function value and the computed moving average is then evaluated:

$$\text{convergence} = |\bar{f}\text{obj} - f\text{obj}^{it}| \leq \tau \tag{1.86}$$

where τ represents the tolerance for the convergence check. This criterion is applied only after the 10^{th} iteration, ensuring that the optimization process has sufficient time to evolve from its initial state and preventing premature termination due to early fluctuations. Compared to alternative methods, this approach generally leads to faster convergence while maintaining solution quality. Another common approach for defining convergence criteria is to monitor the maximum change in the element density vector between successive iterations.

1.6.1 Benchmark problems

The following sections present a set of standard benchmark problems used to assess and validate the performance of the developed TO tool. These test cases are commonly employed in the literature and span a variety of structural configurations. The objective is to compare the numerical results obtained from the in-house optimization framework with those from the commercial ANSYS Workbench software, focusing on stiffness maximization under a volume constraint. Additionally, to demonstrate the effectiveness of the implemented extrusion and maximum size constraints, two supplementary test cases are included: the plate and the cylinder. For these two last cases, we employed the extrusion constraint as an effective strategy to generate rib layout design.

All problems are defined using the following material properties:

- $E = 200$ GPa
- $\nu = 0.3$
- $\gamma = 7850$ kg/m^3

It is important to note that only a qualitative comparison was performed, as ANSYS does not provide access to key quantitative details such as the actual objective function value, maximum displacement, or stress levels.

3D Cantilever Beam

The 3D cantilever beam consists of a rectangular domain with dimensions $L = 2$ m, $H = 1$ m, and $W = 1$ m. The left face is fully fixed, and a downward point load of 1000 N is applied at the bottom of the opposite face, as shown in Fig. 1.9. The finite element discretization employs the SOLID185 element type, with a total of 64k elements

used in the mesh. The optimization problem is defined as stiffness maximization under a volume constraint of 30%.

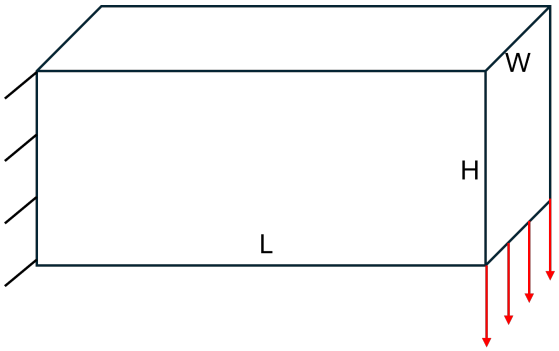
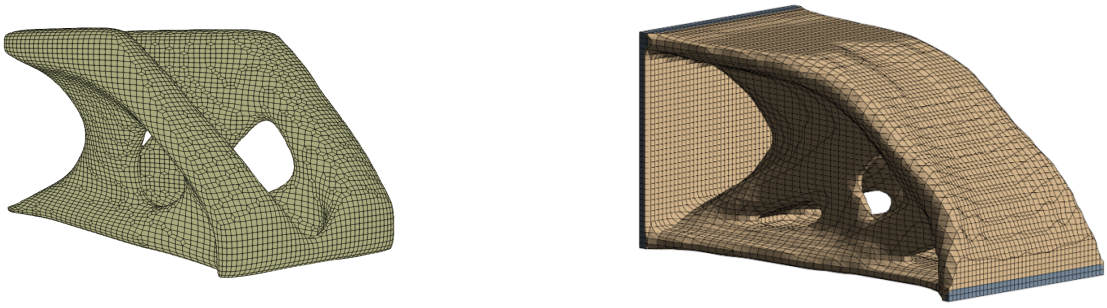


Figure 1.9: 3D Cantilever Beam.

The optimized topology obtained using the in-house code and ANSYS Workbench exhibit noticeably different material distributions. While both solutions maintain structural connectivity, the Workbench solution retains more material around the fixed end and the area where the load is applied. This leads to a more robust and stable structure, especially near critical regions (see Fig. 1.10b). In contrast, the solution from the developed tool tends to reduce material more aggressively, resulting in a slightly simpler topology with fewer redundant paths (see Fig. 1.10a). This may suggest that the density filter and optimization settings in the custom implementation favor global stiffness over local reinforcement near constraints.



(a) (b)

Figure 1.10: (a) In-house TO: $n_{iter.} = 188$, $t = 2.14$ h, $C_{opt} = 40441$ Nm; (b) ANSYS TO: $n_{iter.} = 51$, $t = 9.30$ min.

3D MBB Beam

The 3D MBB beam has dimensions $L = 3$ m, $H = 0.5$ m, and $W = 0.5$ m. A vertical point load of 1000 N is applied at the middle of the top surface.

The bottom left edge is constrained in both vertical and horizontal directions, while the bottom right edge in vertical and normal directions (constraint normal to zy-plane to prevent rigid body motion), as shown in Fig. 1.11. The finite element discretization employs the SOLID185 element type, with a total of 125k elements used in the mesh. The optimization aims to maximize stiffness using only 30% of the volume.

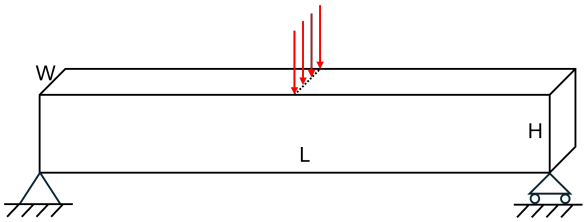
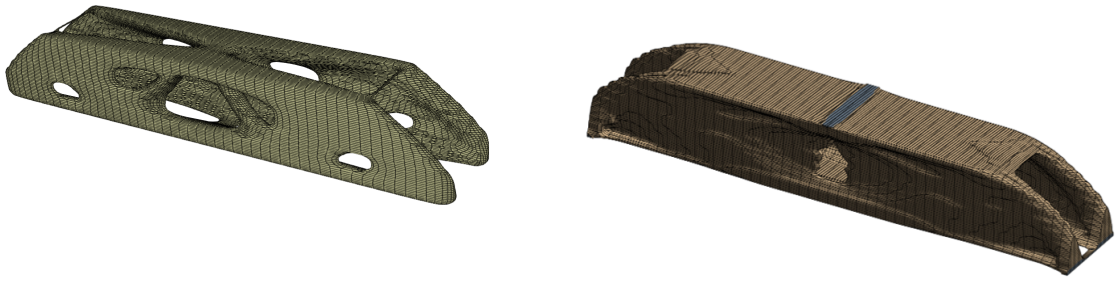


Figure 1.11: 3D MBB Beam.

In the MBB beam case, the topology generated by the in-house optimizer contains a higher number of holes compared to the ANSYS Workbench outcome (see Fig. 1.12a). The Workbench solution tends to retain material more consistently around the boundary conditions and load applications, forming a more continuous structure (see Fig. 1.12b). In contrast, the in-house solution removes material even in some regions where constraints are applied, resulting in a more perforated design with less emphasis on local reinforcement. This suggests that Workbench favors preserving structural continuity near critical regions, while the in-house approach prioritizes global stiffness more aggressively, potentially at the expense of local compliance.



(a) (b)

Figure 1.12: (a) In-house TO: $n_{iter.} = 200$, $t = 6.37$ h $C_{opt} = 341$ Nm; (b) ANSYS TO: $n_{iter.} = 29$, $t = 10.39$ min.

3D L-Bracket

This problem features an L-shaped bracket with horizontal (L) and vertical (H) legs of length 1 m, respectively, and a thickness (W) of 0.33 m. A vertical point load of 1000 N

is applied at the end of the free horizontal leg, while the top face of the vertical leg is fixed, as shown in Fig. 1.13. The finite element discretization employs the SOLID185 element type, with a total of 40k elements used in the mesh. The optimization problem focuses on stiffness maximization with a 30% volume constraint.

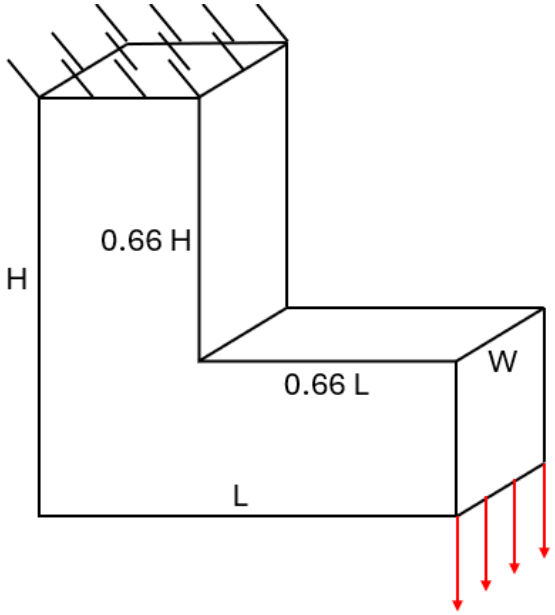
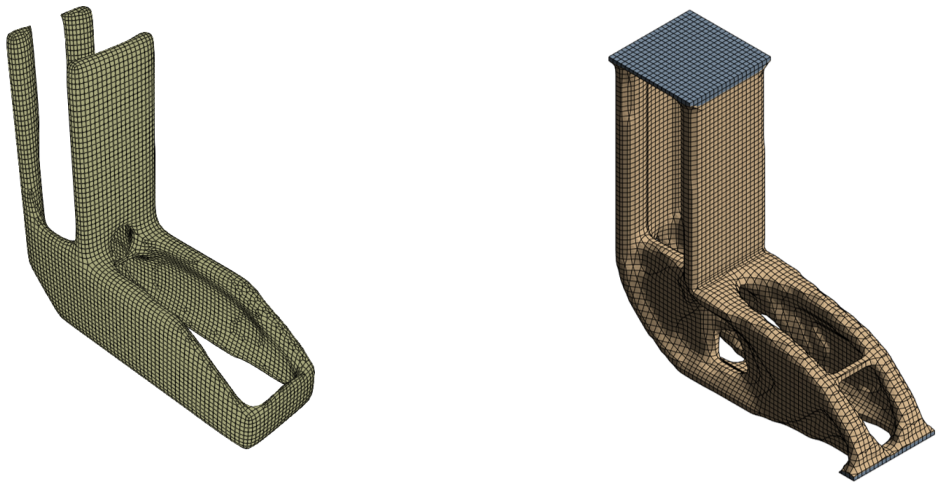


Figure 1.13: 3D L-Bracket.

In the L-Bracket case, the solution generated by the in-house optimizer appears denser overall, with fewer internal holes and more solid regions (see Fig. 1.14a). The material tends to remain well-distributed across the bracket, resulting in a stiffer-looking structure. In contrast, the ANSYS Workbench solution forms a more refined topology, including thinner members and finer features, especially near the load application zone (see Fig. 1.14b). This suggests that Workbench may better capture the need for local reinforcements, while the in-house tool favors broader material retention and simpler connectivity patterns.

Plate

The analyzed plate is 5×5×0.4 m. A vertical load $F_z = -404$ kN is applied on the right side, while the opposite side is fixed, as illustrated in Fig. 1.15b. The mesh consists of 16 layers along the z direction (see Fig. 1.15a), with 3 layers of elements forming Ω_{ndd}, which has a thickness of $t_{\text{ndd}} = 0.075$ m; a total of 160k elements composes the FE model. The average element size is 0.04 m. The extrusion constraint is enforced in the z direction, and the final material distribution is restricted to 30% of the initial volume Ω_{dd} ($G_v = 0.3$).



(a) (b)

Figure 1.14: (a) In-house TO: $n_{iter.} = 179$, $t = 1.00$ h, $C_{opt} = 2381$ Nm; (b) ANSYS TO: $n_{iter.} = 36$, $t = 3.39$ min.

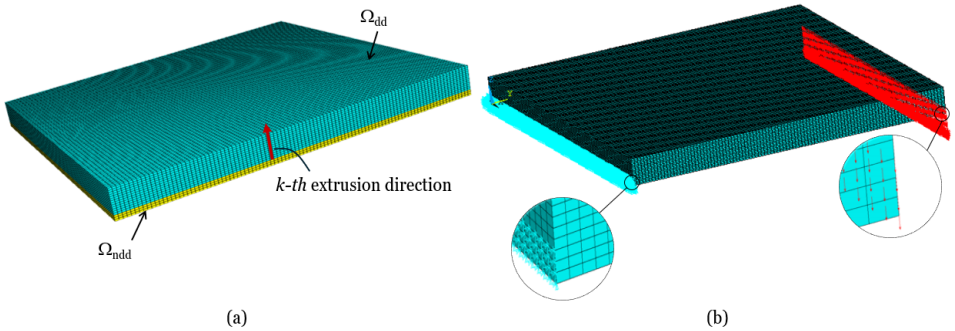


Figure 1.15: Plate FE model: (a) Ω_{dd} and Ω_{ndd} domains; (b) applied loads and displacement constraints.

The optimized plate without considering manufacturing constraints is shown in Fig. 1.16. The initial mass is 78.50 tons. The initial structural compliance, C_{init}, is $4.460 \cdot 10^8$ Nm calculated by setting the element densities in Ω_{dd} equal to the prescribed volume fraction, while the final compliance, C_{opt}, is $9.301 \cdot 10^7$ Nm. The final mass is 33.86 tons. The optimization process primarily removes material near the central region, promoting thinner members at the edges and forming complex curved structures.

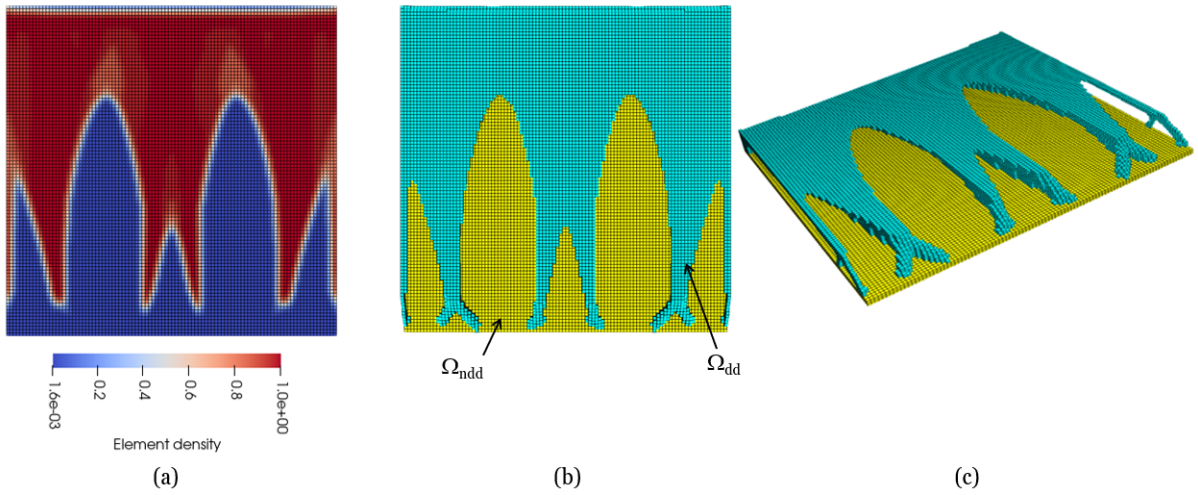


Figure 1.16: Plate TO results: (a) element density plot; (b) top view ($\tilde{\rho} > 0.5$); (c) isometric view ($\tilde{\rho} > 0.5$).

When manufacturing and volume constraints are introduced, the final structural compliance C_{opt} increases to $11.166 \cdot 10^7$ Nm, but a less complex and more manufacturable design is obtained, as shown in Fig. 1.17. In this case, the extrusion constraint prevents the formation of cavities parallel to the plate surface, leading to a clear rib layout. Additionally, the maximum size constraint with $r_{\max} = 0.24$ m ensures that bulk material is split, preventing excessive material concentration and limiting member sizes within the acceptable range.

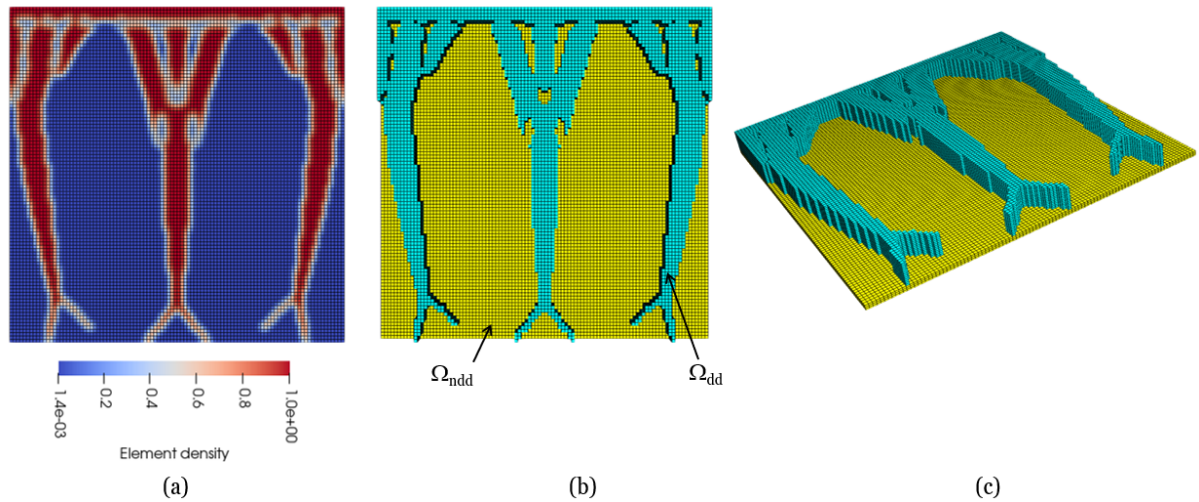


Figure 1.17: Plate TO results with extrusion and max size constraints: (a) element density plot; (b) top view of optimized rib ($\tilde{\rho} > 0.5$); (c) isometric view of optimized rib ($\tilde{\rho} > 0.5$).

The convergence history for the plate case is illustrated in Fig. 1.18. The Fig. 1.18a corresponds to the standard TO case, where the normalized objective function rapidly decreases within the first few iterations before stabilizing. The Fig. 1.18b represents the

case where manufacturing, volume constraints, and an adaptive move-limit strategy are applied. Here, the convergence behavior exhibits a more gradual decrease in compliance, reflecting the additional constraints imposed on the design space. The smoother transition observed in the constrained case indicates enhanced numerical stability, which helps to prevent abrupt changes in the material distribution and ensures a manufacturable final design.

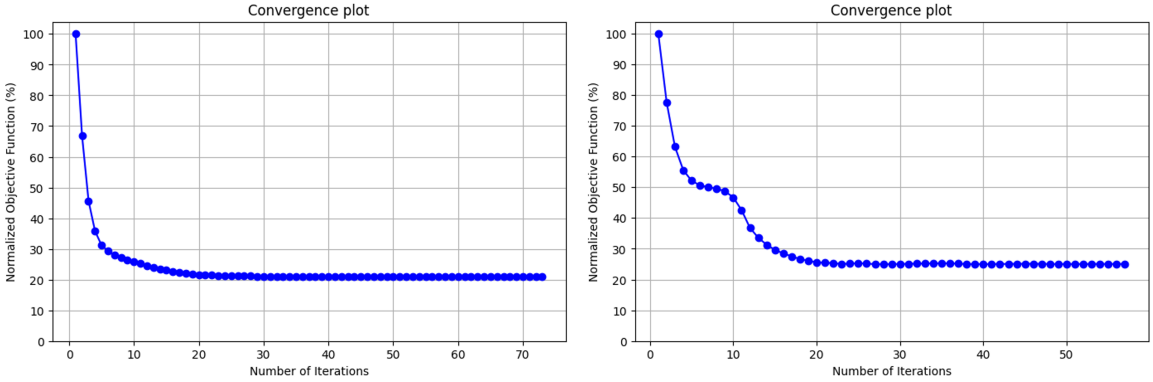


Figure 1.18: Plate TO convergence plot: (a) normalized objective function with no manufacturing constraints; (b) normalized objective function with manufacturing constraints and decreasing m_l strategy.

Cylinder

The analyzed cylinder has an inner radius of 2 m, an outer radius of 2.5 m, and an axial length of 10 m. A net force $F_x = 1$ MN is applied to the inner middle section, directed along the x axis, while both ends are fixed (see Fig. 1.19b). The FE mesh consists of 10 layers along the radial direction (see Fig. 1.19a), with 3 layers of elements forming Ω_{ndd}, which has a thickness of $t_{\text{ndd}} = 0.15$ m. In total, the FE model comprises 160k elements, and the average mesh size is 0.076 m. The extrusion constraint is applied in the radial direction, and the final material distribution is restricted to 30% of the initial Ω_{dd} volume ($G_v = 0.3$).

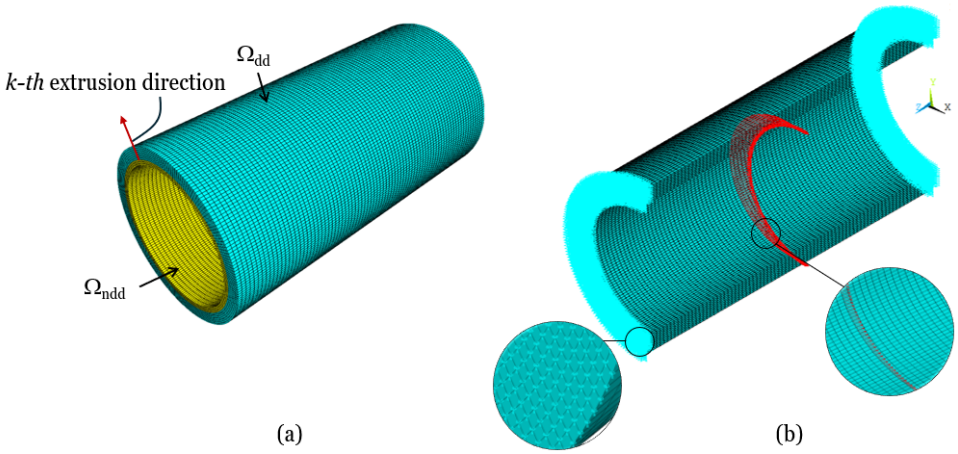


Figure 1.19: Cylinder FE model: (a) Ω_{dd} and Ω_{ndd} domains; (b) applied loads and displacement constraints.

The optimized cylindrical structure without manufacturing constraints, depicted in Fig. 1.20, has an initial mass of 555 tons. The initial compliance is $C_{init} = 9.477 \cdot 10^8$ Nm, computed by setting the element densities in Ω_{dd} equal to the prescribed volume fraction. After optimization, the compliance reduces to $C_{opt} = 5.314 \cdot 10^8$ Nm, with a final mass of 274 tons. Due to the symmetric geometry, loads, and displacement constraints, the design exhibits a rib-like structure, but significant material concentration occurs in the central region.

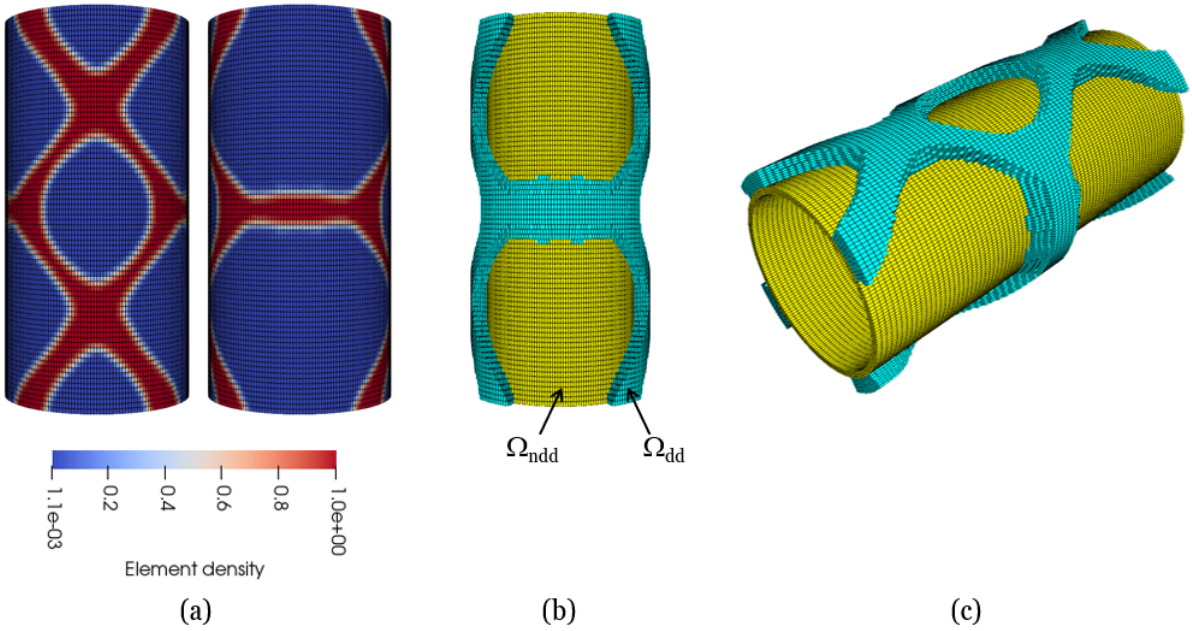


Figure 1.20: Cylinder TO results: (a) element density plot; (b) top view ($\tilde{\rho} > 0.5$); (c) isometric view ($\tilde{\rho} > 0.5$).

Introducing manufacturing and volume constraints slightly increases compliance to

$C_{opt} = 5.617 \cdot 10^8$ Nm, as shown in Fig. 1.21. The extrusion constraint prevents excessive material buildup at the center by introducing radial holes, improving manufacturability while preserving structural efficiency. The maximum size constraint, set to $r_{\max} = 0.45$ m, ensures a balanced material distribution and limits oversized structural members, resulting in an optimized yet practical design.

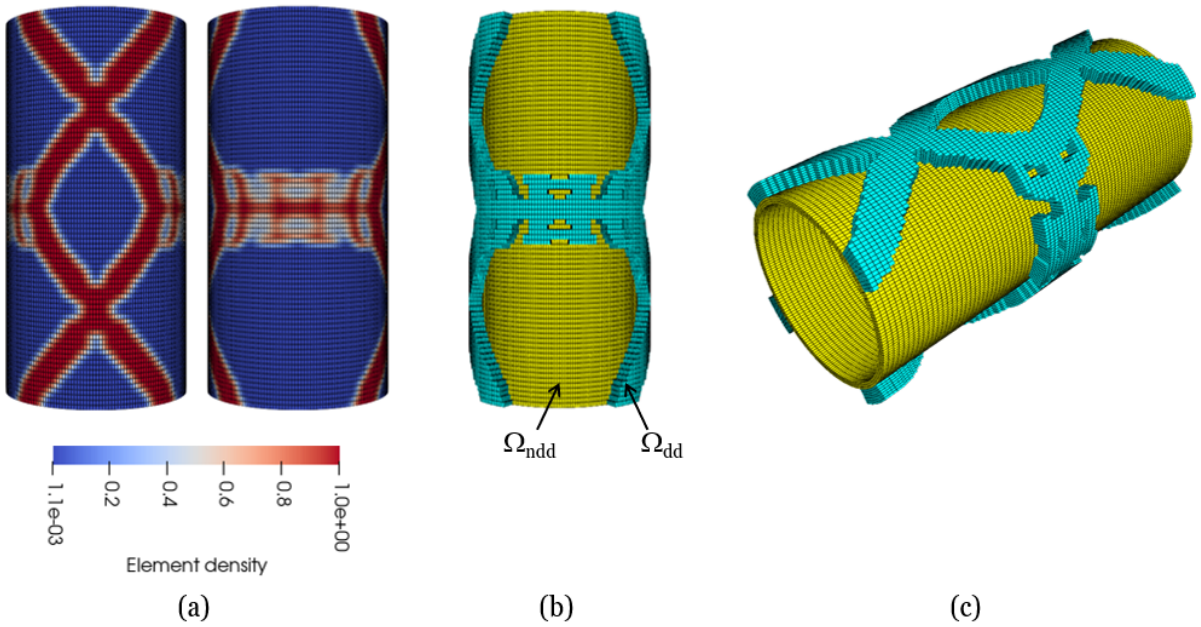
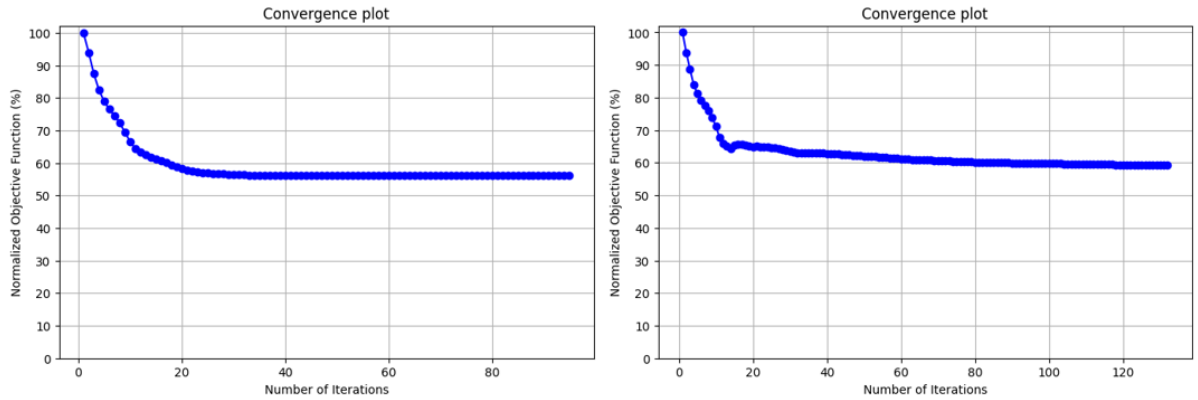


Figure 1.21: Cylinder TO results with extrusion and max size constraints: (a) element density plot; (b) top view of optimized rib ($\tilde{\rho} > 0.5$); (c) isometric view of optimized rib ($\tilde{\rho} > 0.5$).

The convergence history for the cylinder case is shown in Fig. 1.22. The Fig. 1.22a corresponds to the standard TO approach, where the normalized objective function decreases steeply in the early iterations before stabilizing. The Fig. 1.22b presents the case where manufacturing and volume constraints and an adaptive move limit strategy are applied. The introduction of these constraints leads to a different convergence behavior: the objective function initially follows a similar rapid decrease but then exhibits a more gradual decline in later iterations. This behavior reflects the influence of manufacturing constraints, which limit abrupt changes in material distribution and enforce manufacturability at the cost of slightly slower convergence. The gradual refinement observed in the constrained case demonstrates the effectiveness of the move limit strategy in stabilizing the optimization process and ensuring a practical final design.



(a) (b)

Figure 1.22: Cylinder convergence plot: (a) normalized objective function with no manufacturing constraints; (b) normalized objective function with manufacturing constraints and decreasing m_l strategy.

1.6.2 Conclusions on benchmark problems

A qualitative comparison between the in-house developed TO code and ANSYS Workbench reveals distinct differences in the resulting structures. While both approaches aim to maximize stiffness under a volume constraint, their outcomes differ due to algorithmic strategies and implementation details.

ANSYS Workbench employs either OC or SQP algorithms, which, coupled with its built-in filtering and constraint handling techniques, tends to preserve more material near load and boundary condition regions. This often results in more refined and complex geometries, featuring a larger number of thin structural connections. The convergence of ANSYS TO is generally faster, which can be advantageous for quick evaluations.

In contrast, the in-house method, based on the MMA algorithm, exhibits more aggressive material removal strategies. The solutions tend to be sparser in regions where structural connections may still contribute to local stiffness, occasionally removing material even around constrained zones. Consequently, the generated topologies appear simpler and more perforated in comparison.

Overall, while ANSYS excels in speed and refined structural detail, the in-house code provides superior flexibility and transparency in post-processing, making it a valuable tool for detailed structural assessment and further customization.

1.7 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [30, 31] is a population-based algorithm designed to solve global optimization problems, both constrained and unconstrained. It operates by iteratively improving a set of candidate solutions, known as particles, based on a defined objective function. Each particle explores the search space by updating its position and velocity at each iteration.

The movement of a particle is influenced by two main factors: its own best-known position (personal best) and the global best-known position found by the swarm (or particles). By combining individual experience with collective knowledge, the swarm is guided toward promising regions of the search space. As the algorithm progresses, particles converge toward optimal or near-optimal solutions through this cooperative behavior. The advantage of this algorithm is that it doesn't require gradient information to advance towards the optimal solution, but it suffers from slow convergence when a large number of DVs are employed.

The PSO algorithm was employed to explore alternative structural layouts for the large cylindrical magnet presented in chapter 3. Its use is motivated by the fact that the problem at hand involves a shape optimization task, where the objective is to identify the most effective reinforcement configuration without relying on gradient information to advance the optimization.

1.7.1 PSO: Analytical Formulation

The algorithm that describes the PSO uses a velocity vector to update the current position of each particle in the swarm; the velocity vector is updated based on the "memory" gained by each particle, conceptually resembling an autobiographical memory, as well as the knowledge gained by the swarm. Thus, the position of each particle in the swarm is updated based on the social behavior of the swarm which adapts to its environment by returning to promising regions of the space previously discovered and searching for better positions over time. Numerically, the position x of a particle i at iteration $k+1$ is updated as:

$$x_i^{k+1} = x_i^k + v_i^{k+1}\Delta t \tag{1.87}$$

where v_i^{k+1} is the particle velocity vector at iteration k, while Δt is the time step set as unit value.

The velocity vector of each particle within the swarm is calculated as:

$$v_i^{k+1} = w\ v_i^k + c_1 r_1 \frac{p_i^k - x_i^k}{\Delta t} + c_2 r_2 \frac{p_g^k - x_i^k}{\Delta t} = w\ v_i^k + c_1 r_1 \left(p_i^k - x_i^k\right) + c_2 r_2 \left(p_g^k - x_i^k\right) \quad (1.88)$$

where:

- r_1 and r_2 are random numbers in the range of 0 and 1;
- p_i^k is the best particle position of particle i at iteration k;
- p_g^k is the global best position in the swarm up to iteration k;
- c_1 and c_2 are, respectively, cognitive parameter and social parameter ($c_1 = c_2 = 2$);
- w is the inertia parameter ($w = 0.9$).

Each particle, which represents a candidate solution of the optimization problem, is characterized having n-dimensional vector position and velocity. The dimensionality is given by the number of DVs used in the optimization.

This gradient-free algorithm is relatively simple, as it only requires the evaluation of the particle positions (defined by the set of DVs) and the corresponding velocity vectors. making it particularly suitable for problems where there is no explicit or direct relationship between the DVs and the objective function, such as in black-box optimization scenarios

The velocity update, as shown in Eq. 1.88, incorporates both individual learning and social interaction mechanisms. Specifically, the cognitive parameter c_1 controls the tendency of a particle to return to its own best-known position (personal experience), while the social parameter c_2 governs the attraction toward the best known position discovered by the swarm (collective experience). These two components, modulated by inertia parameter w, allow PSO to balance exploration and exploitation across the design space and during the optimization.

In general, PSO algorithm is based on the following steps:

1. Initialize a set of particles positions x_i^0 and velocities v_i^0 randomly distributed throughout the design space bounded by specified limits;
2. Evaluate the objective function values f^k using the design space positions x_i^k;
3. Update the optimum particle position p_i^k at current iteration k and global optimum particle position p_g^k;

4. Update the position of each particle using its previous position and updated velocity vector through Eq. 1.88;

5. Repeat steps 2–4 until the stopping criteria is met. The basic stopping criteria is defined based on the number of iterations reached.

The initialization phase is critical to ensure that the optimization process converges toward a global optimum rather than being trapped in local minima. In particular, it is important to distribute the swarm homogeneously across the entire design space, especially in problems where the location of the global optimum is unknown. Common strategies to enhance initial coverage include Latin Hypercube Sampling (LHS), Sobol sequences, and other low-discrepancy methods, which help to avoid clustering and ensure a uniform distribution of particles [32].

1.7.2 Constrained PSO

The basic PSO algorithm does not inherently consider constraints on the DVs. However, in most engineering problems, the DVs are bounded by physical or design limitations, such as dimensional constraints. These constraints are often expressed as inequalities, for example a thickness parameter limited between 20 mm and 50 mm.

Moreover, the standard PSO algorithm does not account for constraints related to the system response, such as stress, displacement, or frequency requirements for instance. These types of constraints on the outputs of the analysis are equally critical in engineering design and must be handled through additional constraint-handling techniques to ensure acceptable solutions.

Boundary Handling

Many practical optimization problems impose lower and upper bounds on DVs. Several strategies have been developed to ensure particles remain within these bounds [33, 34]:

- Modify personal/global best update: in this simplest method, the particle's position and velocity remain unchanged, but infeasible particles are not selected as personal or global bests. Their objective function evaluations can be skipped or assigned a high penalty value.

- Reposition infeasible particle: this approach directly adjusts particles that move outside the feasible bounds:

  - *Nearest method:* particles exceeding the bounds are clamped to the boundary:

$$x_i^{k+1} = \begin{cases} x_{\text{up}}, & \text{if } x_i^{k+1} > x_{\text{up}} \\ x_{\text{low}}, & \text{if } x_i^{k+1} < x_{\text{low}} \end{cases}$$

  - *Pbest method:* particles are returned to their last feasible personal best position.

  To prevent the particle from immediately exiting the feasible region again due to its "momentum", the velocity is often modified:

  - Adsorb: reset the velocity to zero: $v_i^{k+1} = 0$
  - Deterministic back: reverse and scale the velocity to bring the particle back: $v_i^{k+1} = -\lambda v_i^{k+1}$, where $\lambda \in [0, 1]$ (commonly $\lambda = 0.5$)
  - Random back: a stochastic variation of the deterministic back approach

- Preventing infeasibility: instead of repositioning after violation, these techniques constrain the velocity update such that the particle does not leave the feasible domain:

  - Hyperbolic method: normalizes the velocity to ensure the next position is within bounds.
  - Random forth: projects the particle to a random point between its current position and the nearest boundary.

Constraint Handling

To ensure that the optimization remains within the feasible design space, constraint-handling strategies must be implemented. A common and effective approach is the Penalty Function technique, which increases the objective function value for infeasible solutions [35].

In this method, a penalty term is added to the objective function when one or more constraints are violated. The penalized objective function f_{pen}^k can be expressed as:

$$f_{pen}^k = f^k + \sum_{j=1}^{m} \lambda_j^k \max(0, g_j^k(\mathbf{x}))^2 \tag{1.89}$$

where:

- f^k is the original objective function;

- $g_j^k(\mathbf{x}) \leq 0$ are the constraint functions:
- λ_j^k are the penalty coefficients;
- m is the total number of constraints.

The penalty coefficients λ_j^k must be carefully selected: if they are too small, the algorithm may still accept infeasible solutions; if too large, the search may be overly restricted and converge prematurely. This technique allows unconstrained optimization algorithms like PSO to be adapted for constrained problems in a straightforward and flexible way.

An easier approach to account for constraints within the optimization process involves normalizing both the objective and constraint functions with respect to reference values, without the need of defining the parameters λ_j^k. This not only simplifies their integration but also ensures that the functions, often expressed in different units, are brought to a comparable scale. Additionally, the objective function can be penalized whenever a constraint is violated:

$$\bar{f}_{pen}^k = \frac{f^k}{f_{ref}} + \sum_{j=1}^{m} \left(\frac{g_j^k(\mathbf{x})}{g_{j,ref}} \right)^p \tag{1.90}$$

Eq. 1.90 was implemented in this work due to its simplicity and its ability to easily scale all parameters, such as objective and constraint functions.

Furthermore, in this work a combination of repositioning and velocity adjustment techniques was implemented to maintain feasibility of all particle positions during the optimization (deterministic back and nearest method). Additionally, LHS was used in the initialization phase to ensure a homogeneous distribution of particles across the feasible design space.

The implementation, fully developed within the APDL (ANSYS Parametric Design Language) environment, is provided in appendix .5.

1.7.3 Simply supported beam optimization

The effectiveness of the PSO algorithm was assessed by applying it to a benchmark case, namely a simply supported beam. The beam is subjected to a 1000 N force, as shown in Fig. 1.23. The goal of this optimization problem is to minimize the beam's weight while ensuring that the stress does not exceed the allowable limit. The DVs are the beam's cross-sectional dimensions, which must be determined to achieve the minimum weight. The width (W) and height (H) of the beam are constrained to be no less than 10 mm,

and the maximum stress anywhere in the beam must not exceed 200 MPa. The beam is assumed to be made of steel ($\gamma = 7850$ kg/m^3) with a Young's modulus of 200 GPa. Furthermore, the FE model of the beam was developed using BEAM188 elements.

The analyzed problem is taken from [36]. The reference solution is:

- $W = 13.24$ mm
- $H = 29.16$ mm
- $\sigma_{max}^{vM} = 199.83$ MPa
- $V = 386100$ mm^3

All the results obtained are shown in Table 1.1.

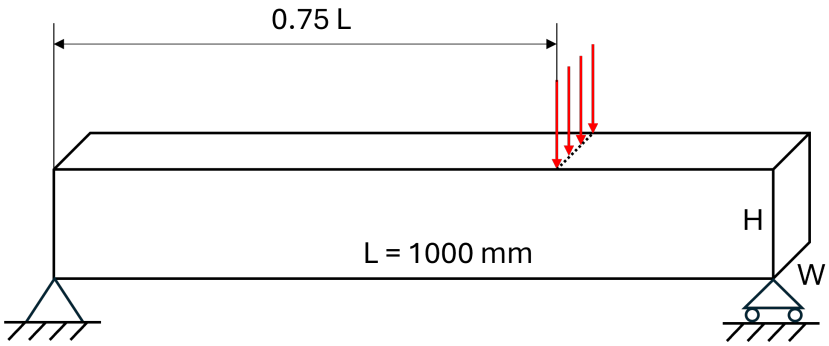


Figure 1.23: Simply supported beam model.

Table 1.1: Beam optimization results for different settings of maximum iterations and particle number using PSO algorithm.

Sol. n°	Max iter.	Particle n°	H [mm]	W [mm]	V [mm^3]	$\sigma_{\text{max,eq}}$ [MPa]
1	100	10	26.60	15.54	413307	197.9
2	10	100	30.82	11.86	365351	193.2
3	10	10	33.94	10.19	345935	185.3
4	10	10	36.15	10.08	364289	165.2
5	10	10	36.27	10.46	379214	158.2
6	20	10	33.33	10.07	335748	194.3
7	10	20	31.77	12.44	395076	173.3
8	30	2	26.76	17.70	473771	171.5
9	30	30	32.65	10.96	357860	186.2
10	10	200	32.17	10.90	350813	192.7

Comparing the reference solution with those shown in Table 1.1, it can be observed that the obtained results are generally consistent with the reference in terms of both stress and volume, but they exhibit slight variations in the cross-sectional dimensions.

For instance, the best solutions (e.g., Solutions 3, 6, and 10) achieve volumes in the range of 335000–351000 mm^3 with stresses close to the allowable limit, while others (e.g., Solution 8) result in larger volumes but lower stresses. In all cases, the maximum stress remains below the 200 MPa threshold.

It should be emphasized that the final solutions obtained are always slightly different from the reference one and from each other. This is due to the stochastic nature of the employed optimization algorithm, which introduces variability in the convergence path and final outcome.

2 Nuclear Fusion: A brief introduction

The global demand for energy continues to rise steadily, driven by rapid industrial development in many emerging countries. Currently, the majority of energy production relies on fossil fuels such as oil, natural gas, and coal. These sources, however, are finite and represent the primary contributors to environmental pollution and greenhouse gas emissions. As a result, there is an urgent need to transition toward sustainable and cleaner energy alternatives. Among these, nuclear fusion represents a promising solution, offering the potential for large-scale, low-emission energy production that could significantly reduce our dependence on fossil fuels.

2.1 Fundamentals of fusion energy

Nuclear fission is the most widely used nuclear reaction for energy production, with many operational reactors around the world. Fission offers several advantages, including the absence of direct gas emissions. However, it also presents notable drawbacks, such as the long-term management of radioactive waste and the potential for weaponization.

In contrast, nuclear fusion (the process that powers the sun and stars) offers a fundamentally safer and more sustainable approach. Although fusion reactions also produce radioactive by-products, the associated risks are considerably lower. Neutron irradiation can induce radioactivity in structural materials, and tritium (one of the fuels used) has a relatively short half-life of about 12 years. Nevertheless, careful selection of structural materials can mitigate these effects, and the environmental impact of fusion waste is generally lower and shorter-lived than that from fission [37].

Fusion fuel is another significant advantage: deuterium is widely available, while tritium can be breed from the interaction between neutron and lithium within the reactor. Additionally, the fusion reaction naturally stops if the fuel supply is interrupted. Fusion systems typically operate at very low pressure minimizing the risk of catastrophic failure or explosion. Moreover, only a small quantity of tritium is used, further enhancing operational safety.

Although nuclear fission and fusion are fundamentally different processes, both are governed by Einstein's mass-energy equivalence principle:

$$E = (m_r - m_p) \cdot c^2 \tag{2.1}$$

where m_r is the mass of the reactants, m_p is the mass of the products, and c is the speed of light in vacuum. Even a small mass difference between reactants and products can result in the release of a tremendous amount of energy, due to the large value of c^2.

In fission, a heavy atomic nucleus (such as uranium-235 or plutonium-239) is split into two lighter nuclei upon absorption of a neutron, releasing energy and additional neutrons that can sustain a chain reaction. In fusion, by contrast, two or more light atomic nuclei, typically isotopes of hydrogen, combine to form a heavier nucleus. This process also releases energy, and in most practical applications, produces high-energy neutrons as a by-product.

A plasma is generally required for fusion reactions, and it is classified as fourth state of matter. While the solid, liquid, and gaseous states are electrically neutral, consisting of atoms with positively charged nuclei surrounded by negatively charged electrons, plasma differs significantly. When a gas is heated to extremely high temperatures, the thermal energy causes the atoms to collide with such intensity that electrons are stripped from their nuclei. This results in a highly ionized gas composed of free electrons and ions that move independently. Unlike neutral gases, plasma can conduct electricity and respond strongly to electric and magnetic fields.

In stars, the formation of plasma and subsequent fusion reactions occur naturally due to the high temperatures (between 5 and 15 millions degrees Celsius) and immense gravitational forces, which both heat and confine the plasma, enabling sustained fusion. Replicating this process on Earth, however, presents a major technological challenge. To achieve fusion under terrestrial conditions, a device must be capable of heating the fusion fuel to temperatures exceeding 100 million degrees Celsius and simultaneously confining the resulting plasma long enough for the fusion reactions to occur efficiently.

To achieve nuclear fusion, one must overcome the Coulomb barrier, the electrostatic repulsion between positively charged atomic nuclei. Since like charges repel, fusion does not occur easily under normal conditions. However, as the temperature of the fuel increases, the kinetic energy of the particles also increases. At temperatures on the order of 100 million degrees Celsius, nuclei move fast enough that, during collisions, some are able to get close enough for the strong nuclear force to overcome the repulsive Coulomb force, allowing fusion to occur.

The probability that a fusion reaction will happen during a collision between two

particles is quantified by the cross-section. This cross-section depends strongly on the particles' impact energy: the higher their energy, the greater the probability that they will overcome the Coulomb barrier and successfully fuse. Therefore, increasing particle energy not only enhances the rate of collisions but also the chance that each collision results in fusion.

The most promising fusion reactions involve light nuclei, especially the isotopes of hydrogen: deuterium (D or ^{2}H) and tritium (T or ^{3}H). The key fusion reactions involving these isotopes include the following:

$$\mathrm{D + T \rightarrow {}^4He + n + 17.6\ MeV}$$
$$\mathrm{D + D \rightarrow {}^3He + n + 3.3\ MeV}$$
$$\mathrm{D + D \rightarrow T + H + 4.0\ MeV}$$
$$\mathrm{D + He \rightarrow {}^4He + H + 18.3\ MeV}$$

In these fusion reactions, helium (also referred to as an alpha particle) and neutrons are among the primary products. A significant portion of the energy released (approximately 80% for D-T reaction) is carried away as kinetic energy by the neutrons. This high-energy neutron flux plays a crucial role not only in the overall energy yield but also in breeding tritium when interacting with lithium. The total energy released in each reaction is shown on the right-hand side of the respective equations above and is distributed among the reaction products. In physics, this energy is commonly measured in eV, where 1 eV corresponds to approximately 11,605 K (based on the Boltzmann constant).

Deuterium, one of the key fuels for fusion, is abundant in seawater (around 30 grams per cubic meter). Its nucleus consists of one proton and one neutron, giving it roughly twice the mass of regular hydrogen (which has only a single proton). Lithium (Li), another essential element in the fusion fuel cycle, is naturally found in the Earth's crust (about 30 parts per million) and in smaller quantities in oceans. It is primarily extracted from salt lakes and plays a key role in tritium production: when lithium absorbs high-energy neutrons from the fusion reaction, it undergoes a nuclear reaction that produces tritium.

Tritium, required for the most favorable fusion reaction with deuterium, consists of one proton and two neutrons. Although it is radioactive, its beta radiation is relatively low-energy and cannot penetrate human skin, posing a health risk only if inhaled or ingested.

Another fusion product is helium, a stable and non-toxic noble gas. It consists of two protons and two neutrons and is completely inert, posing no environmental or health hazard.

Among the various possible fusion reactions, the deuterium-tritium (D+T) reaction is considered the most promising and practical, as it has the highest cross-section—meaning the highest probability of occurrence—at relatively low ignition temperatures. This is illustrated in the diagram in Fig. 2.1.

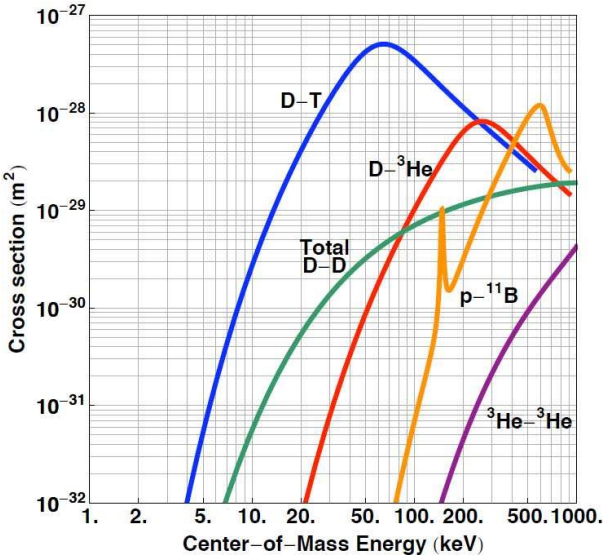


Figure 2.1: Fusion cross-section vs. temperature for different reactions. The D–T reaction peaks at the lowest temperature [4].

In the D–T reaction, helium nuclei (alpha particles) and neutrons are produced. The helium nuclei, being positively charged, are confined by the magnetic fields within the plasma. In contrast, neutrons are neutral and thus are unaffected by magnetic fields—they escape the plasma and are absorbed by the surrounding walls of the fusion device. This energy transfer from neutrons to the walls, in future fusion reactors, will be converted into heat, which will be subsequently used to generate steam that drives a turbine connected to an alternator, producing electricity.

As mentioned earlier, the neutrons generated in the D–T reaction are also absorbed by a specially designed component of the reactor known as the breeding blanket. This blanket contains lithium, which reacts with the neutrons to produce tritium and helium via the following nuclear reaction:

$$^{6}\mathrm{Li} + \mathrm{n} \rightarrow {}^{4}\mathrm{He} + \mathrm{T} + 4.86\ \mathrm{MeV}$$

This reaction is essential for maintaining the tritium supply required for the D-T fusion process.

The main challenge associated with the D–T reaction is the use of tritium, a radioactive isotope that is not naturally abundant. As such, tritium must be bred within the

reactor itself. Due to its radioactive nature and the high-energy neutrons produced, advanced shielding and remote handling systems are required to ensure the safe operation of the reactor.

2.2 Experimental Devices for Magnetic Confinement

To sustain fusion reactions, effective plasma confinement is crucial. There are three main approaches:

- **Magnetic confinement:** plasma is enclosed within a vacuum chamber, and an appropriate configuration of external magnetic fields and or a field produced by current induced in the plasma prevent contact with the vessel walls. Charged particles can interact with magnetic fields, while neutral particles escape from plasma;
- **Inertial confinement:** a small sphere of Deuterium and Tritium (also known as pellet) is subjected to micro explosions using high-energy laser or ion beams. The energy is transferred onto the sphere's surface, which evaporates immediately, and the fuel is compressed and heated; in this way the very high-density condition of the plasma is achieved, and fusion reactions take place;
- **Gravitational confinement:** plasma is compressed to a high density and temperature by a gravitational field due to the enormous mass. In this way, the matter is so dense and hot that the nuclei cannot repel each other, because the force of gravitational field exceeds the electrostatic repulsion.

The two most important fusion devices that use magnetic confinement are Tokamak and Stellarator. Both of them, represented in Fig. 2.2 confine plasma in a toroidal (doughnut-shaped) vacuum chamber. The main differences between the two devices are:

- **Tokamak:** utilizes both external magnetic coils (producing a toroidal field) and an induced plasma current (producing a poloidal field) to create a helical magnetic field. This system is pulsed in nature and susceptible to plasma disruptions;
- **Stellarator:** achieves the helical magnetic field configuration entirely through externally twisted magnetic coils, without relying on plasma current. This allows for steady-state operation and greater stability.

From a physical point of view, the Tokamak benefits from toroidal symmetry, which simplifies modeling using the Grad–Shafranov equations. In contrast, Stellarators are inherently three-dimensional and require more complex analysis and optimization to achieve good confinement properties.

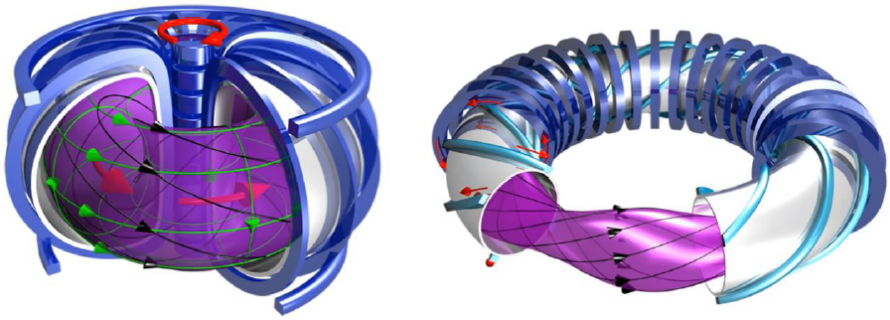


Figure 2.2: A Tokamak (on the left) and Stellarator (on the right) representation.

2.3 Lawson Criterion

The Lawson Criterion defines the minimum conditions under which a fusion reactor can achieve a net positive energy output. It is typically expressed as the triple product:

$$n \cdot T \cdot \tau_E \geq 1.2 \cdot 10^{21} \text{ m}^{-3} \text{ keV s} \tag{2.2}$$

where n indicates the density of particles, τ_E indicates the confinement time (i.e., time required to completely dissipate the plasma energy if the sources that are feeding it are interrupted) and T the temperature.

To meet the Lawson Criterion, all three parameters must simultaneously reach sufficiently high values. While current devices can achieve high values of individual parameters, achieving all three together remains a major technological challenge.

Improvements to τ_E in magnetic confinement devices can be achieved by:

- Increasing the magnetic field strength, which enhances particle confinement, but at the cost of increasing the overall stress on the magnet system;
- Increasing the size of the reactor, which increases the path length for particle escape, but at the cost of increasing the overall reactor size.

Meeting the Lawson Criterion is a crucial step toward making fusion a sustainable and safe energy source for the future.

2.4 The role of Renaissance Fusion

Renaissance Fusion (RF) is a French start-up based in Grenoble, founded in 2020, with the following core objectives [38]:

- Demonstrate an innovative High-Temperature Superconductor (HTS) manufacturing process with potential applications in medical imaging and simplified stellarator design;
- Construct the first high-field (10 T) HTS-based stellarator featuring thick, flowing liquid metal walls;
- Deliver fusion electricity to the grid by 2030s.

Among experimental fusion devices, tokamaks are currently the most advanced, with stellarators following closely behind. In recent years, interest in stellarators has grown significantly due to increased investments and their inherent ability to operate in steady-state, making them promising candidates for continuous energy generation.

However, stellarator fusion reactors development presents several key challenges:

- Neutron-induced radioactivity, which damages solid reactor components;
- Large reactor size required for achieving net energy gain;
- Complex coil geometries, which increase manufacturing and assembly difficulty.

It is worth noting that the first two challenges are common to tokamaks devices, whereas the third is specific to stellarators due to their intrinsically three-dimensional magnetic field topology and non-axisymmetric coil design.

To address these challenges, RF is developing a compact, high-field stellarator using:

- HTS magnets capable of producing strong magnetic fields, thus reducing reactor size;
- Liquid metal walls, which absorb neutron radiation and mitigate material degradation;
- Modular coil designs, which simplify manufacturing and allow for on-site assembly.

Traditional stellarator coil production involves a series of intricate steps:

1. HTS, electrical, and mechanical layers are deposited on tapes by multiple machines;
2. These tapes are twisted helically into cables;
3. Cables are wound onto non-planar forms to create complex 3D coils;

4. The coils are precisely installed around the vacuum vessel with sub-millimeter accuracy.

RF proposes a novel, streamlined approach to coil fabrication:

- Multiple HTS layers are deposited directly onto a cylindrical ((or nearly cylindrical) substrate;
- Current paths are defined via laser grooving, which etches the HTS film with sub-millimeter precision;
- A numerically controlled process rotates the cylinder while the laser scans it axially, achieving accurate, repeatable patterns.

This approach eliminates the need for bending or forming cables and enables the creation of highly precise and complex current distributions. The same process can be replicated across all modules, which are later assembled to complete the Stellarator's magnetic coil system.

2.5 Topology Optimization in Nuclear Fusion

In recent years, Topology Optimization (TO) has emerged as a powerful tool in structural design, particularly in fields such as as aerospace, automotive, and biomedical engineering [39, 40, 41]. The TO strength lies in the ability to generate efficient, lightweight, and mechanically robust structures by systematically distributing material within a defined design space, often finding non-intuitive solutions that would be difficult to achieve through traditional design approaches. The solutions generated by TO are often highly complex and not directly compatible with traditional manufacturing methods, especially for large scale problems. As a result, the integration of manufacturing constraints into the TO workflow has become a critical area of research [19, 20]. Without such constraints, that limit the space of admissible solutions, the optimized geometry may be difficult or even impractical to fabricate, requiring significant manual effort and time to convert the result into a clean engineering solution.

In the context of nuclear fusion, TO showed its potential for the development of the FFHR-c1 helical fusion reactor [42]; here, researchers applied TO to redesign the coil support structures. This approach led to a significant weight reduction of more than 25% compared to conventional designs. Subsequent static and seismic analyzes confirmed the structural integrity of the optimized supports under both operational and earthquake-induced loads. TO has also been utilized to determine optimal placements

and orientations of permanent magnets in Stellarators [43]; this method aims to simplify coil designs while maintaining effective plasma confinement, potentially reducing manufacturing complexity and enhancing performance. Researchers have also employed thermal TO to design plasma-facing components, such as the divertor [44]. By optimizing the distribution of the high thermal conductivity material within these components, the designs achieve improved thermal performance and structural resilience under extreme operating conditions. Further research applied TO to design tungsten-copper structures for plasma-facing component applications [45]. This approach improves the thermal and mechanical performance of components exposed to high heat fluxes, ensuring durability and efficiency in fusion reactor environments. The integration of automated parameter, shape, and topology optimization techniques has been also explored in the design of divertors [46]. These methods aim to optimize the geometry and material distribution to handle power exhaust effectively while maintaining structural integrity and manufacturability.

Despite these promising applications, most of the existing research on TO in nuclear fusion has focused either on relatively localized components (e.g., divertors, plasma-facing materials) or on global coil support structures with limited constraints. A notable gap remains in the application of TO to where manufacturing requirements must be also addressed.

In particular, the use of extrusion constraints, commonly applied to enforce manufacturability by maintaining a constant material distribution along a specified direction, has not been extended to handle multiple subdomains. Moreover, to the best of the author's knowledge, this approach has never been applied in the context of structural optimization for nuclear fusion magnets. This limitation often leads to designs that represent an "average" structural response across the entire domain, potentially overlooking localized features that are critical for mechanical performance and realistic manufacturability.

The work presented in chapter 4 addresses this gap by introducing and applying a multiple extrusion constraint approach within the TO framework. This method enhances the optimizer's ability to capture structurally relevant features, ultimately producing more realistic and manufacturable reinforcement concepts for stellarator fusion magnets, like those under design by RF.

3 Parametric Modeling for 10 T, 8.5 m bore Stellarator magnet reinforcements

This chapter focuses on the development of advanced parametric modeling techniques aimed at optimizing the structural reinforcement systems of the large scale stellarator under design by RF. The goal is to provide reliable mechanical reinforcements under high EM loading conditions while reducing structural mass and complying with manufacturability constraints. Parametric modeling enables a flexible and efficient framework for exploring different reinforcement configurations, allowing rapid design iteration and optimization, but more restriction on the final outcomes come into the design compared to the topological approach.

In this study, the limit stress σ_{lim} is set to 900 MPa and is compared with the maximum equivalent von Mises stress within the stress region (Fig. 3.3 for each optimization case. The exact material for the reinforcement structures has not yet been finalized. For the purposes of this analysis, a material with properties similar to those used in ITER was assumed (SS316 LN), exhibiting high strength of approximately 1000 MPa at cryogenic temperatures near 4 K [47]. The magnet under consideration will operate at around 20 K; therefore, a yield strength of 1000 MPa for the structural reinforcements was adopted. The assumed stress limit represents a preliminary value with a relatively low safety margin within the elastic range of the material. In subsequent design stages, this limit may be reduced to account for additional operational loads and to ensure an adequate safety margin.

No displacement and strain constraints were imposed at this stage. This choice was intentional, as the primary aim was to explore the design space and evaluate the structural performance under a less restrictive or over constrained formulation. Introducing only a stress constraint allowed the optimizer greater flexibility in modifying the reinforcement topology. Subsequent analyses will evaluate the resulting displacements and strains to confirm that they remain acceptable and to quantify their influence on the magnetic

field accuracy resulting from magnet deformation, as well as magnet safety. If these analyses show a significant impact, displacement and strain limits will be enforced through additional penalty terms in the objective function.

To achieve these goals, a combination of FEA and stochastic optimization techniques, particularly PSO, is employed. This hybrid approach allows the exploration of large design spaces while efficiently converging toward structurally sound and material efficient solutions. The optimization targets include minimizing total structural weight, while constraining the maximum stress and the geometrical complexity to be manufacturable.

The following parametric models have been developed and analyzed:

- Minimum vessel thickness (sec. 3.3);
- Poloidally variable thickness (sec. 3.4);
- Double-wall shell with internal poloidal ribs (sec. 3.5);
- Double-wall shell with internal beam reinforcements (sec. 3.6).

Each optimization problem addressed, can be stated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^p \\ \text{s.t.} \quad & \sigma_{max} \leq \sigma_{lim} \\ & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \ldots, n \end{aligned} \tag{3.1}$$

where m_{real} is the mass of the optimized structure at a given iteration, σ_{max} represents the maximum stress within the stress evaluation region (see Fig. 3.3), m_{ref} is the reference mass, set to 1500 t, and the penalty function used for the stress constraint is a simple power law depending on the parameter p, as shown in Eqs. 3.1, which penalizes the objective function when the stress limit is exceeded.

The vector of DVs ($\mathbf{x} = x_1, x_2, ..., x_n$) contains the geometric parameters being optimized, which are highlighted for each analyzed scenario. Each design variable is constrained between lower and upper bounds, x_i^L and x_i^U, respectively, ensuring that the solutions remain manufacturable. Finally, all the results reported in the subsequent sections correspond to the best global objective function achieved by the swarm.

3.1 Magnet System Overview

The magnet system design relies on four large cylindrical superconducting magnets arranged symmetrically. Each magnet has a diameter of approximately 8.5 m, with a

maximum axial length of 16.6 m at outer radius and a minimum axial length of 3.6 m at inner radius, as shown in Fig. 3.1. The magnet system is designed to generate a magnetic field of up to 10 T at the plasma axis.

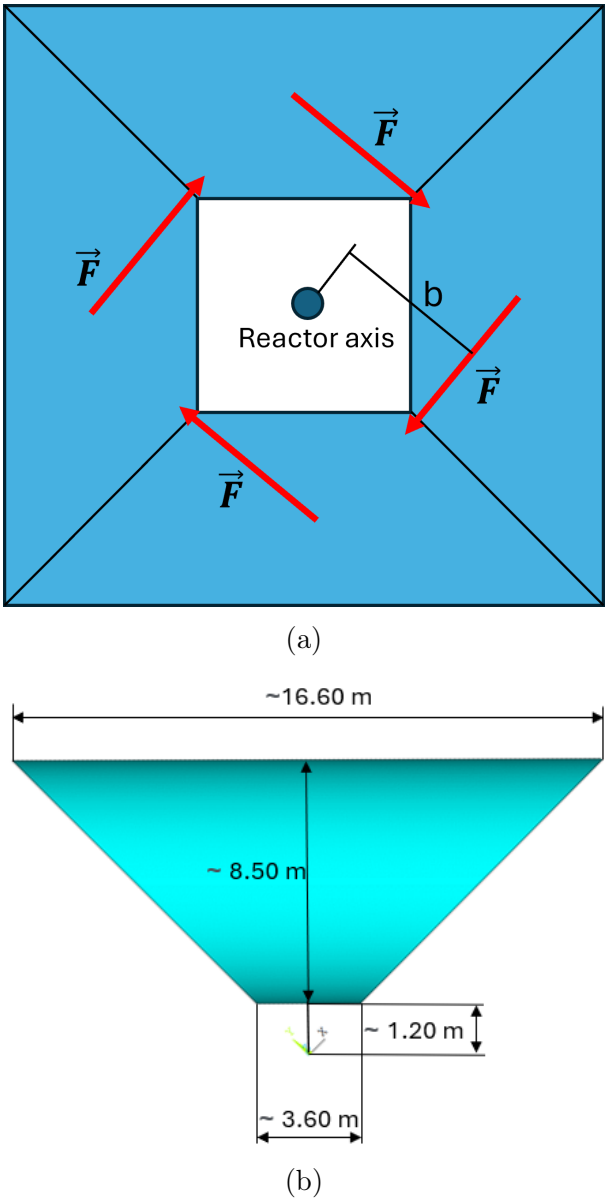


Figure 3.1: (a) Magnet system overview, (b) Top view and main dimensions of one magnet.

During the design process of the magnet system, EM loads were computed to estimate the mechanical demands on the structural supports. For each magnet, the EM forces were found to be not radially symmetric and not directed toward the geometric center (see Fig. 3.1a). As a consequence, the coil loads are not self-balanced, meaning that the net force resulting from summing all coil contributions does not strictly vanish. This imbalance arises from preliminary calculations performed at an early stage, using numerical models that did not yet implement the constraint of globally self-balanced force distri-

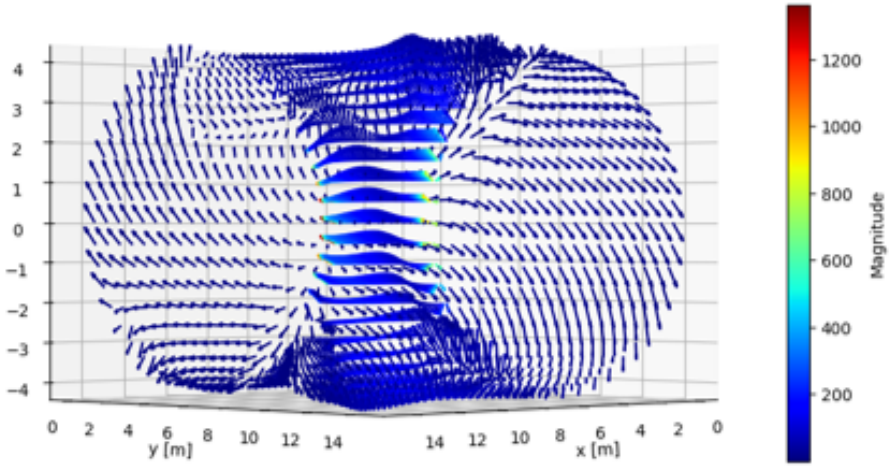


Figure 3.2: EM loads plot [MPa].

butions. Although the absence of enforced equilibrium conditions affects the absolute accuracy of the computed EM forces, it does not invalidate the exploration of structural concepts. Therefore, the present results still provide a representative mechanical loading scenario that enables development and testing of the structural optimization framework. As the numerical modeling for EM loads computation is refined in future iterations, with the inclusion of force-balancing constraints, the methodology developed in this work can be directly applied to updated load cases with minimal adjustment.

One of the most challenging aspects of the magnet design is the presence of extremely high EM forces generated during operation. The peak values of these forces can reach locally up to 1200 MPa, posing significant mechanical stresses on the coil structure and its reinforcement components (see Fig. 3.2). As a result, the mechanical integrity of the reinforcement system surrounding the coils must be carefully engineered to withstand these forces without excessive stresses.

The magnet reinforcements are supposed to envelope the magnets. More specifically, the magnets will occupy a small portion of the reinforcement structure and will be placed closer to the plasma, in order to minimize the magnetic field error.

3.2 FE model

Fixed displacement constraints are imposed on the lateral ends of the magnet structure to replicate the increased stiffness due to mechanical junctions and to prevent unrealistic torsional deformations resulting from the non-self-balanced EM loads. The evaluation of mechanical stress is restricted to a region located 0.5 m from both magnet ends, enabling the analysis to focus on a representative structural volume while avoiding artificially high stress peaks caused by boundary constraints and localized high peak EM loads (see

Fig. 3.3).

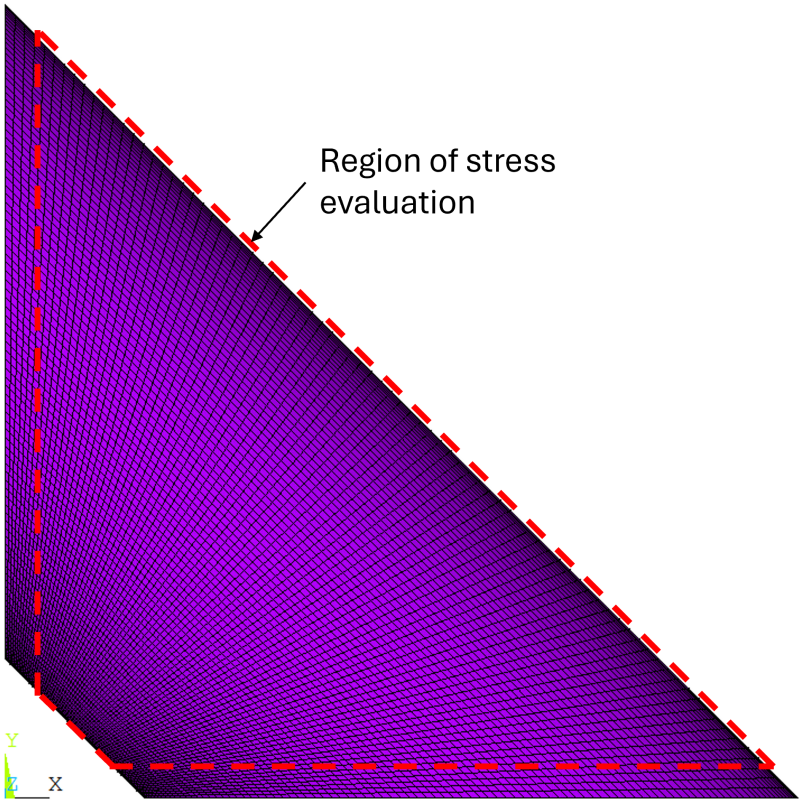


Figure 3.3: Region of stress evaluation (0.5 m from both magnet ends).

Each FE model is built using SHELL181 elements, with BEAM188 elements included in the final model that features internal beam reinforcements. The EM loads, expressed as pressure distributions, are interpolated and applied to the inner surface of the magnet reinforcement using ANSYS APDL. In this study, material plasticity is not considered, and the material behavior is assumed to be linearly elastic even when stresses exceed the yield strength of SS316LN. Consequently, in cases where the stress overcomes the yield limit, the resulting displacements may be significantly underestimated.

Thermal effects and self-weight are neglected in this study, as the EM loads dominate the structural response by several orders of magnitude. This assumption allows the optimization framework to concentrate solely on the structural behavior under EM loading, which is the primary design goal in this context.

Differently from chapter 4, the magnets were not modeled using the homogenization technique. This assumption was adopted to simplify the analysis and focus on the structural performance and optimization methodology, rather than on detailed material heterogeneity.

3.3 Minimum Vessel Thickness

The first parametric study focuses on identifying the minimum allowable thickness of the magnet structure. The primary design variable is thickness t of the cylindrical magnet casing.

The optimization problem can be expressed as:

$$\begin{aligned} \min_{\mathbf{t}} \quad & m = \frac{m_{\text{real}}}{m_{\text{ref}}} + \left(\frac{\sigma_{\max}}{\sigma_{\text{lim}}}\right)^3 \\ \text{s.t.} \quad & \sigma_{max} \leq 900 \text{ MPa} \\ & 50 \text{ mm} \leq t \leq 2000 \text{ mm} \end{aligned} \tag{3.2}$$

where m is the total structural mass and σ_{max} is the maximum von Mises stress measured in the designated evaluation zone.

Regarding the PSO settings, the population is composed of 10 particles (referred as n_{PSO} in Table 3.1) and the maximum number of iterations is 10, 20 and 30 (referred as $\text{it}_{\max}$ in Table 3.1) in order to ensure that the optimization algorithm reached a stable optimum point.

All the results obtained for the constant thickness configuration are summarized in Table 3.1, while in Fig. 3.11a is shown the outcome of the 2^{nd} case. The optimization of the single thickness magnet reinforcement model was performed to identify the minimum uniform thickness that satisfies the imposed mechanical constraint. Three evaluations were conducted under different conditions. In the 1^{st} case, a thickness of 675 mm resulted in a total mass of 1714 t, with a maximum stress of 712 MPa and a maximum displacement of 21 mm. In the 2^{nd} case, the optimized thickness was reduced to 544 mm, achieving a structural mass of 1381 t, with a maximum stress of 898 MPa and a displacement of 30 mm. A 3^{rd} evaluation yielded a nearly identical thickness of 543 mm, maintaining the same mass (1381 t) and satisfying the stress constraint (σ_{max} = 898 MPa). These results indicate that, for a constant-thickness reinforcement model, the optimal solution that minimizes structural mass while complying with the stress limit is achieved with a thickness of 543 mm.

Table 3.1: Optimization results for constant-thickness reinforcement model.

Case	t [mm]	m [t]	σ_{max} [MPa]	u_{max} [mm]	$\text{it}_{\max}$	n_{PSO}
1	675	1714	712	21	10	50
2	544	1381	898	30	20	50
3	543	1381	898	30	30	50

While the design meets the stress constraint, the resulting thickness is relatively large,

which may cause difficulties in the manufacturing process. The structure is therefore likely to be over-conservative in regions experiencing lower stresses, leading to excess material usage and reduced design efficiency.

This outcome serves as a reference for more advanced parametric models introduced in the following sections, which aim to improve material distribution, reduce overall structural thickness and mass.

3.4 Poloidally variable thickness

In this model, the reinforcement thickness is allowed to vary along the poloidal direction to better accommodate the local structural demands imposed by the EM loads. The DVs are defined as twelve thickness values, t_i, each associated with a specific angular segment of the magnet cross-section (see Fig. 3.4). The optimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{t}} \quad & m = \frac{m_{\text{real}}}{m_{\text{ref}}} + \left(\frac{\sigma_{\max}}{\sigma_{\text{lim}}}\right)^p \\ \text{s.t.} \quad & 50 \text{ mm} \leq t_i \leq 2000 \text{ mm}, \quad i = 1, \ldots, 12 \\ & \sigma_{\max} \leq \sigma_{\text{lim}} \end{aligned} \tag{3.3}$$

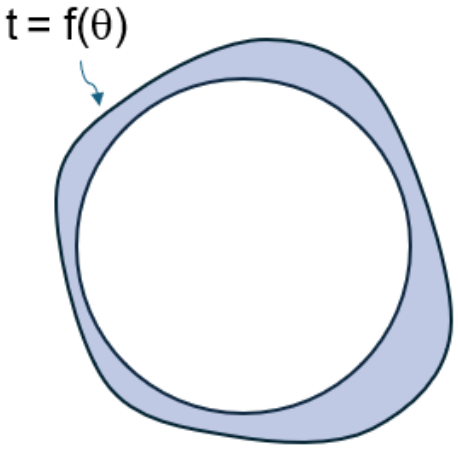


Figure 3.4: Cross-section of the poloidally variable thickness model.

To ensure a smooth variation of thickness and to avoid abrupt changes that may lead to stress concentrations, a Bézier interpolation is applied to connect the t_i values. This interpolation provides a continuous thickness profile normal to the circular cross-section of the magnet.

Two separate optimization runs were performed using a population size of 50 and a maximum of 50 iterations. The first run adopted a penalization exponent $p = 3$, while

the second used $p = 5$, increasing the penalty for stress constraint violation. The number of DVs are 12 for both analyzed cases. All the results obtained for the poloidally variable thickness configuration are summarized in Table 3.2, while in Fig. 3.11b and 3.11c are shown the outcome of the 4^{th} and fifth cases.

In the 4^{th} case ($p = 3$), the optimization resulted in a significantly lower structural mass of 970 t, but at the cost of a maximum stress of 1236 MPa, which exceeded the allowable limit. The corresponding maximum displacement was 54 mm, with local thicknesses ranging from 139 mm to 1211 mm.

In the 5^{th} case ($p = 5$), the optimizer favored a thicker configuration to better comply with the stress constraint. The final mass increased to 1383 t, while the maximum stress slightly decreased to 1179 MPa. The maximum displacement was reduced to 33 mm, and the local thickness varied between 168 mm and 1644 mm.

Table 3.2: Optimization results for poloidally variable thickness configuration.

Case	$t_{min} - t_{max}$ [mm]	m [t]	σ_{max} [MPa]	u_{max} [mm]	$\text{it}_{\max}$	n_{PSO}	p
4	139–1211	970	1236	54	50	50	3
5	168–1644	1383	1179	33	50	50	5

These results highlight the influence of the penalization exponent on the optimization outcome. A higher penalization exponent led to improved stiffness with the stress constraint but at the cost of increased mass. In both cases, however, the final stress values remained above the imposed limit. This suggests that the stress constraint was not strictly enforced within the optimization framework. When the penalty is not sufficiently steep, the optimizer may treat the stress limit as a soft constraint, balancing it against the objective, rather than as an absolute restriction. As a result, the optimizer prioritized overall performance while tolerating small violations of the stress limit. In addition, the limited number of iterations have further restricted the optimizer to approach a feasible solution, suggesting that further iterations may be required to achieve full constraint satisfaction. Furthermore, in both cases the maximum thickness is still relatively large (more then 1 m), posing difficulties in its manufacturing.

3.5 Double-wall shell with internal poloidal ribs

The double-wall shell model introduces a more structurally efficient reinforcement configuration by combining an Inner Shell (IS), an Outer Shell (OS), and internal Poloidal Ribs (PRs) that connect the two shells.

Initially, the OS was defined with a fixed racetrack shape, selected to facilitate structural integration. Similarly, the PRs were constrained to follow this racetrack geometry,

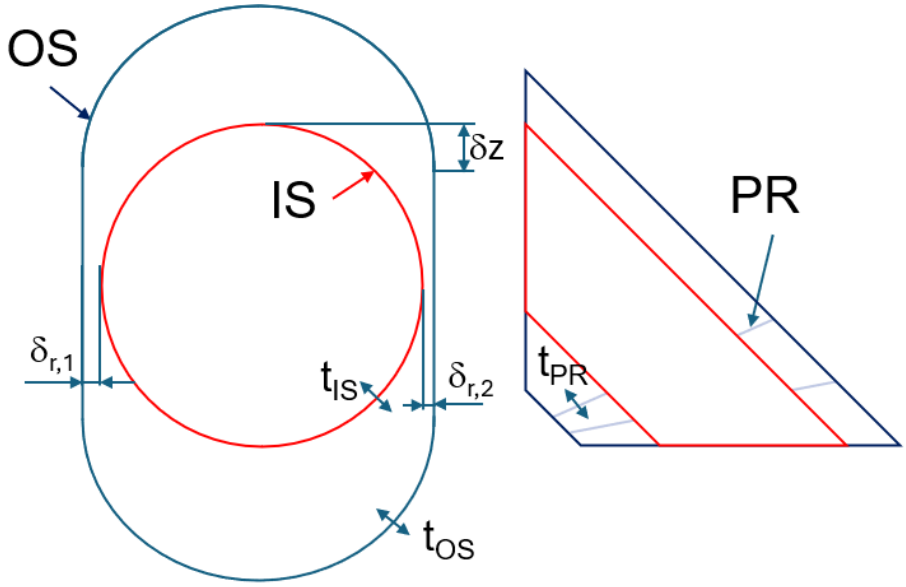


Figure 3.5: Representation of double wall with poloidal ribs like racetrack shape.

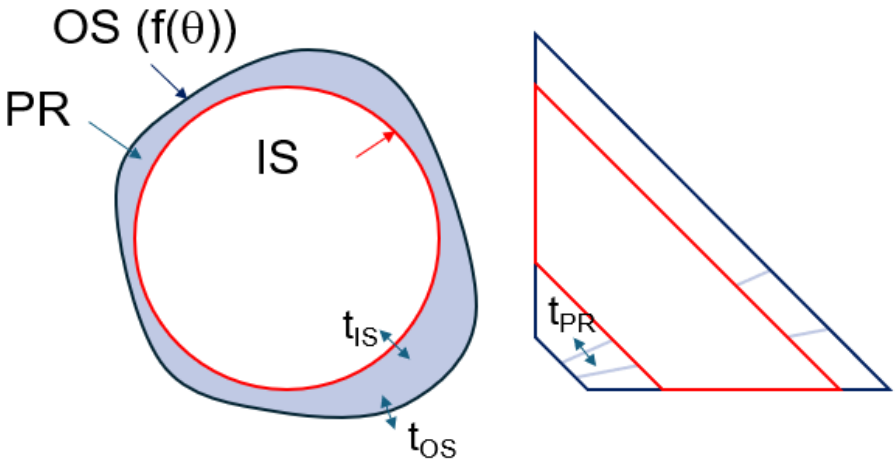


Figure 3.6: Representation of double wall with poloidal variable ribs model.

regularly distributed along the poloidal direction to transfer loads between the IS and OS (see optimization problems 3.4 and 3.5, and Fig. 3.5). These ribs act as stiffeners, enhancing the structural resistance and allowing for overall thickness reduction.

In a subsequent refinement of the model, the geometric constraints on the OS and PRs were relaxed. Inspired by the variable thickness strategy used in the previous sec. 3.4, the shape of the OS, as well as of the PRs, were parameterized using control points and Bézier curves, allowing for smooth and continuous geometric variations along the poloidal direction (see optimization problems 3.6 and 3.7, and Fig. 3.6).

The following optimization problems were formulated for the double-wall shell reinforcement configuration:

- IS-OS-PRs$_{\text{racetrack}}$:

$$\begin{aligned} \min_{\mathbf{t},\delta} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^3 \\ \text{s.t.} \quad & \sigma_{max} \leq \sigma_{lim} \\ & 50 \text{ mm} \leq t_{\text{IS}}, t_{\text{OS}}, t_{\text{PR}} \leq 500 \text{ mm} \\ & 80 \text{ mm} \leq \delta_{r,i} \leq 1500 \text{ mm} \\ & -3500 \text{ mm} \leq \delta_z \leq -1000 \text{ mm} \end{aligned} \tag{3.4}$$

- IS-OS-PRs$_{\text{racetrack}}$:

$$\begin{aligned} \min_{\mathbf{t},\delta} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^3 \\ \text{s.t.} \quad & \sigma_{max} \leq \sigma_{lim} \\ & 50 \text{ mm} \leq t_{\text{IS}}, t_{\text{PR}} \leq 300 \text{ mm} \\ & 0 \text{ mm} \leq t_{\text{OS}} \leq 300 \text{ mm} \\ & 80 \text{ mm} \leq \delta_{r,i} \leq 1500 \text{ mm} \\ & -3500 \text{ mm} \leq \delta_z \leq -1000 \text{ mm} \end{aligned} \tag{3.5}$$

- IS-OS-PRs$_{\text{variable}}$:

$$\begin{aligned} \min_{\mathbf{t},\delta} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^3 \\ \text{s.t.} \quad & \sigma_{max} \leq \sigma_{lim} \\ & 50 \text{ mm} \leq t_{\text{IS}}, t_{\text{PR}} \leq 500 \text{ mm} \\ & 0 \text{ mm} \leq t_{\text{OS}} \leq 500 \text{ mm} \\ & 100 \text{ mm} \leq \delta_i \leq 1500 \text{ mm} \end{aligned} \tag{3.6}$$

- IS-OS-PRs$_{\text{variable}}$:

$$\begin{aligned} \min_{\mathbf{t},\delta} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^3 \\ \text{s.t.} \quad & \sigma_{max} \leq \sigma_{lim} \\ & 50 \text{ mm} \leq t_{\text{IS}}, t_{\text{PR}} \leq 300 \text{ mm} \\ & 0 \text{ mm} \leq t_{\text{OS}} \leq 300 \text{ mm} \\ & 100 \text{ mm} \leq \delta_i \leq 1500 \text{ mm} \end{aligned} \tag{3.7}$$

In problems 3.4, 3.5, 3.6 and 3.7, t_{IS}, t_{OS} and t_{PR} are referred to IS, OS and PR thicknesses, respectively. In problems 3.4 and 3.5, δ_z represents the vertical offset between the highest point of the inner IS and the highest point of the straight section of the OS, when viewed in cross-section (see Fig. 3.5). This parameter effectively controls the vertical

positioning and extent of the OS relative to the IS. Meanwhile $\delta_{r,i}$ represent the minimum and maximum radial distances between OS and the machine center. Furthermore, in problems 3.4 and 3.6, a lower nonzero bound is imposed on t_{OS}, whereas in problems 3.5 and 3.7, the lower bound is set to zero to allow the optimizer to explore the possibility of including or excluding an outer shell. Finally, in problems 3.6 and 3.7, δ_i represents the i-th control point for optimizing the OS shape, as well as the PRs; in these last two problems, the control points are used to create the Bézier curve. The PSO was set to 30 iterations and a population size of 30 for each optimization problem.

For manufacturing reasons it is fundamental to place the PRs far enough from each other to allow sufficient space for welding them on both IS and OS. A rule of thumb is to consider the minimum distance between adjacent ribs as:

$$d_{\mathrm{min,PR}} \geq 2 \cdot t_{\mathrm{PR}} \tag{3.8}$$

The maximum number of PRs at each optimization step is determined based on the minimum available axial length required to accommodate them. This available length depends on the axial extent of the OS at its minimum radial position relative to the machine center, as well as on the thickness of the PRs (t_{PR}). The PRs are uniformly distributed between the IS and the OS, spanning the entire axial length of the magnet. First, the minimum available axial length of the OS, denoted as $L_{\mathrm{OS}}^{\mathrm{min}}$, is estimated based on its inner radius $r_{\mathrm{in,OS}}$:

$$L_{\mathrm{OS}}^{\mathrm{min}} = 2 \cdot r_{\mathrm{in,OS}} \cdot \tan(45^\circ) \tag{3.9}$$

The length occupied by each PR, L_{PR}, depends on the minimum distance between ribs $d_{\mathrm{min,PR}}$ and the rib thickness t_{PR}:

$$L_{\mathrm{PR}} = d_{\mathrm{min,PR}} + t_{\mathrm{PR}} \tag{3.10}$$

The maximum number of PRs, $n_{\mathrm{max,PR}}$, is computed as:

$$n_{\mathrm{max,PR}} = \mathrm{int}\left\lfloor \frac{L_{\mathrm{OS}}^{\mathrm{min}}}{L_{\mathrm{PR}}} \right\rfloor \tag{3.11}$$

where $\mathrm{int}\left\lfloor \frac{L_{\mathrm{OS}}^{\mathrm{min}}}{L_{\mathrm{PR}}} \right\rfloor$ is approximated to the nearest lower integer.
The remaining free space along the OS length, L_{free}, is given by:

$$L_{\mathrm{free}} = L_{\mathrm{OS}}^{\mathrm{min}} - n_{\mathrm{max,PR}} \cdot L_{\mathrm{PR}} \tag{3.12}$$

The offset from one of the two ends of the magnet to the first PR, denoted o_{tor}, is defined as:

$$o_{\text{tor}} = \frac{L_{\text{free}}}{2} + 1.5 \cdot t_{\text{PR}} \tag{3.13}$$

The angular position θ_i of each rib i (with $i = 1, \ldots, n_{\text{max,PR}}$) is then calculated geometrically:

$$\theta_i = \tan^{-1}\left(\frac{(o_{\text{tor}} + (i-1) \cdot L_{\text{PR}}) \cdot \cos(45°)}{L_{\text{OS}}^{\text{min}} \cdot \cos(45°) - (o_{\text{tor}} + (i-1) \cdot L_{\text{PR}}) \cdot \cos(45°)}\right) \tag{3.14}$$

All the results obtained for both poloidally variable thickness and OS racetrack shape configurations are summarized in Table 3.3, while in Fig. 3.11d, 3.11e, 3.11f and 3.11g are represented the outcomes from the sixth to the ninth cases.

The optimization results show significant improvements in structural efficiency across the evaluated configurations. Solutions of problems 3.4 and 3.5 demonstrate a substantial reduction in thickness while maintaining acceptable mechanical performance, with both cases reaching the maximum allowable stress of 900 MPa (cases 6^{th} and 7^{th} in Table 3.3). The corresponding maximum displacements are 28 mm and 24 mm, respectively, and the resulting total masses are 1379 t and 1642 t. In the 6^{th} case, the configuration is characterized by a relatively thick IS ($t_{\text{IS}} = 397$ mm), a minimal OS ($t_{\text{OS}} = 50$ mm), and moderately thin PRs ($t_{\text{PR}} = 63$ mm), with a total of 8 PRs ($n_{\text{PRs}} = 8$). The 7^{th} case instead makes more extensive use of the OS ($t_{\text{OS}} = 249$ mm), while reducing the IS thickness to 299 mm and minimizing rib dimensions to 50 mm, using a total of 16 ribs ($n_{\text{PRs}} = 16$).

The solution of problem 3.6 (case 8^{th} in Table 3.3), although still requiring a relatively large IS thickness ($t_{\text{IS}} = 422$ mm), achieves a significant mass reduction to 1122 t compared to the minimum thickness model in sec. 3.3, while also satisfying the stress constraint with a maximum displacement of 34 mm. Here, the OS is entirely removed ($t_{\text{OS}} = 0$ mm), and the ribs are relatively thick ($t_{\text{PR}} = 135$ mm), with a minimal number of only 2 PRs ($n_{\text{PRs}} = 2$), showing a tendency to concentrate structural function in the ribs and IS.

Finally, the solution of problem 3.7 (9^{th} case in Table 3.3) yields a considerable thickness reduction similar to the 6^{th} and 7^{th} cases, reaching the stress limit of 900 MPa and exhibiting a maximum displacement of 24 mm. The resulting total mass is 1457 t, with a thinner OS ($t_{\text{OS}} = 186$ mm), IS ($t_{\text{IS}} = 297$ mm), and intermediate PRs thickness ($t_{\text{PR}} = 73$ mm), supported by 10 ribs ($n_{\text{PRs}} = 10$).

When enforcing an upper bound of 300 mm on the IS thickness, the optimizer favors

Table 3.3: Optimization results for reinforcement systems with internal PRs.

Case	n_{PRs}	t_{IS} [mm]	t_{OS} [mm]	t_{PR} [mm]	m [t]	σ_{max} [MPa]	u_{max} [mm]	$\mathrm{it}_{\mathrm{max}}$	n_{PSO}
6	8	397	50	63	1379	893	28	20	10
7	16	299	249	50	1642	900	24	30	30
8	2	422	0	135	1122	900	34	30	30
9	10	297	186	73	1457	900	24	30	30

configurations with multiple slender ribs, which help distribute the loads more effectively. In contrast, relaxing this constraint allows the optimizer to eliminate the ribs altogether and compensate with a significantly thicker IS. While this reduces design complexity, the resulting large thickness may pose manufacturing challenges and reduce practicality.

3.6 Double-wall shell with internal beam reinforcements

The double-wall shell with internal beam reinforcements represents an alternative structural strategy in which discrete cylindrical beams are introduced between the IS and OS (see Fig. 3.7). In both cases, the DVs include the thicknesses of the IS and OS, the radius of the beam elements ($\mathrm{r}_{\mathrm{beam}}$) and the offset of OS ($o_{\mathrm{OS}}$) w.r.t. IS (here, the OS shape was considered circular for simplicity).

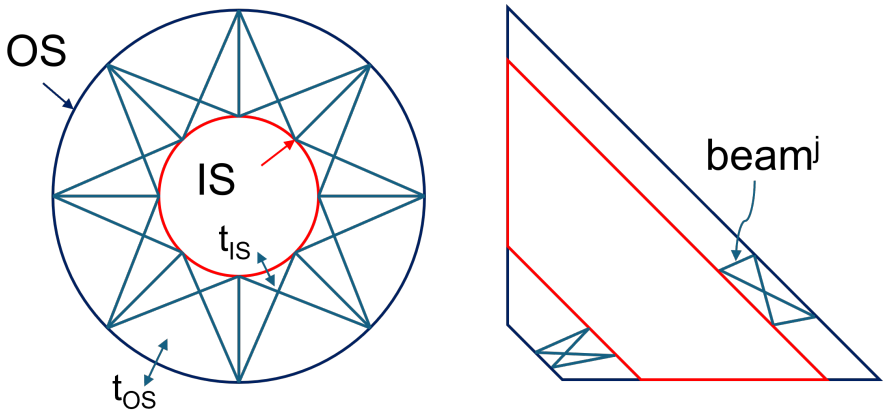


Figure 3.7: Representation of double wall with poloidal ribs and internal beams model.

The optimization problem is defined as following:

$$
\begin{aligned}
\min_{\mathbf{t},r_{\text{beam}},o_{\text{OS}}} \quad & m = \frac{m_{real}}{m_{ref}} + \left(\frac{\sigma_{max}}{\sigma_{lim}}\right)^3 \\
\text{s.t.} \quad & \sigma_{max} \leq 900 \text{ MPa} \\
& 50 \text{ mm} \leq t_{\text{IS}} \leq 300 \text{ mm} \\
& 0 \text{ mm} \leq t_{\text{OS}} \leq 300 \text{ mm} \\
& 100 \text{ mm} \leq o_{\text{OS}} \leq 1500 \text{ mm} \\
& 0 \text{ mm} \leq r_{\text{beam}} \leq 250 \text{ mm}
\end{aligned}
\tag{3.15}
$$

Two optimization runs were evaluated for this configuration due to significant large number of DVs for a gradient free algorithm (1417). The first run (10^{th} case in Table 3.4) was limited to 30 iterations, while the second (11^{th} case in Table 3.4) was extended to 100 iterations. In both optimizations, a particle population of 30 was employed.

All the results obtained for the double-wall shell with internal beam configuration are summarized in Table 3.4, while in Fig. 3.11h and 3.11i are shown the results of cases 10^{th} and 11^{th}.

The results of both optimizations show that the algorithm tends to minimize the contribution of the OS, with its thickness converging toward zero ($t_{\text{OS}} = 12$ mm and 0 mm, respectively). This suggests that the optimizer favors a structurally minimal OS, relying primarily on the IS and beam reinforcements for load resistance. However, both solutions result in relatively high maximum stresses (1371 MPa and 1281 MPa) and displacements (46 mm and 47 mm), exceeding those found in the more efficient poloidal rib configurations. Additionally, the overall structural mass remains considerably high (1946 t and 1932 t), which further indicates that this configuration, while conceptually promising, is less effective in minimizing weight while satisfying stress constraint compared to alternative reinforcement strategies.

Table 3.4: Optimization results for double-wall shell with internal beam reinforcements.

Case	t_{IS} [mm]	t_{OS} [mm]	m [t]	σ_{max} [MPa]	u_{max} [mm]	it_{max}	n_{PSO}
10	300	12	1946	1371	46	30	30
11	299	0	1932	1281	47	100	30

3.7 Uncertainty of PSO results

To quantify whether the cases analyzed and shown in Fig. 3.11 have converged or still exhibit dispersion in their objective function values, a fitness variance metric is employed. This quantity measures the spread of the particles' objective function values within the swarm at a given iteration and provides insight into the optimization process. Although a

predefined convergence criterion was not applied during the optimization, the convergence status of each case was evaluated a posteriori through the analysis of fitness variance and best global objective function history.

From a statistics point of view, the fitness variance is a measure of how spread out a set of values is around their mean (average). The fitness variance is defined as following:

$$\sigma_f^2(t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \left(f_i(t) - \bar{f}(t) \right)^2 \tag{3.16}$$

where N_p is the number of particles in the swarm, $f_i(t)$ is the objective function value of particle i at iteration t, and $\bar{f}(t)$ is the mean fitness value across all particles, given by

$$\bar{f}(t) = \frac{1}{N_p} \sum_{i=1}^{N_p} f_i(t) \tag{3.17}$$

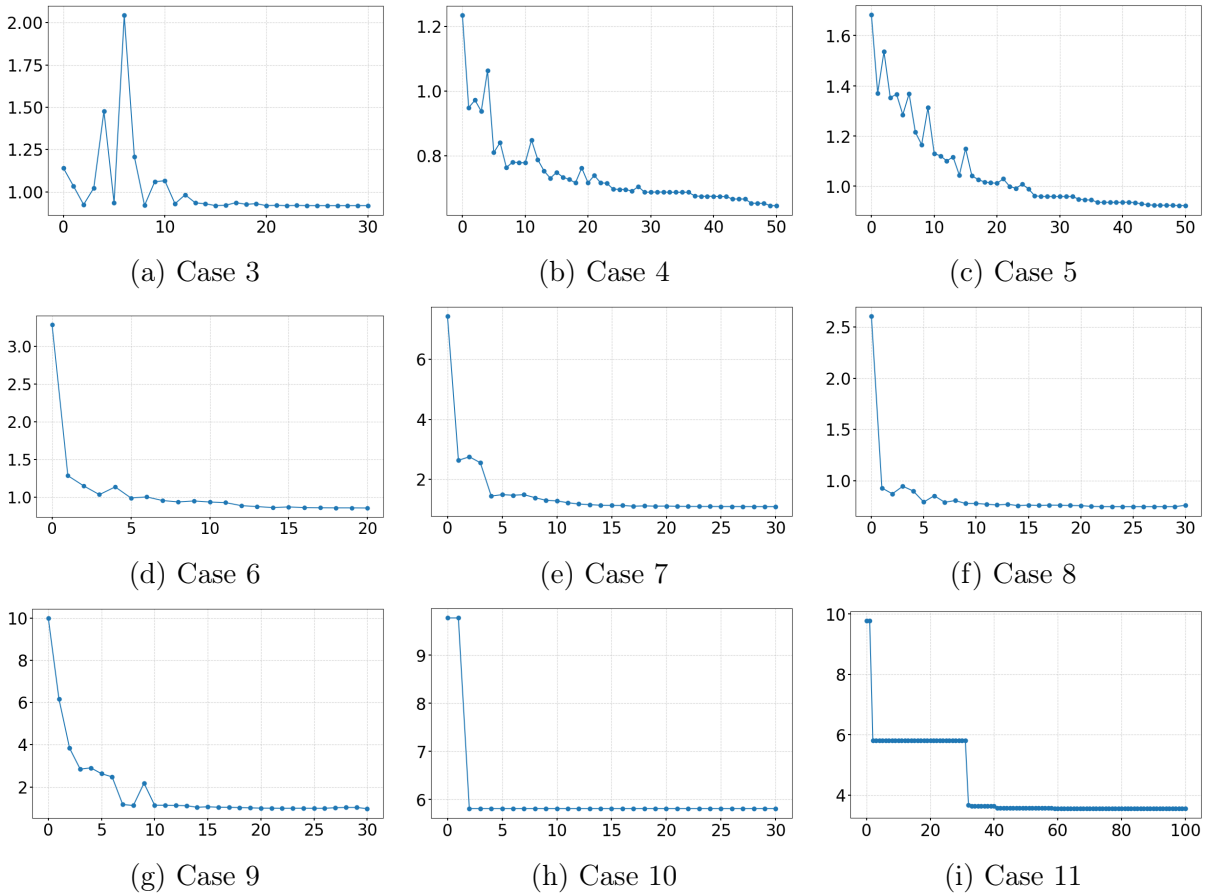
A decreasing $\sigma_f^2(t)$ indicates that the swarm is converging toward a region of similar objective values, while a low or near-zero variance suggests that the particles have reached a stable configuration, potentially corresponding to an optimal or near-optimal solution. The evolution of the fitness variance across particles at each iteration is illustrated in Fig. 3.9, while Fig. 3.8 presents the normalized objective function values, referenced to the best value achieved by the swarm at each iteration.

Cases 3, 6, 8, 9, 10, and 11 exhibit stabilization of the global best objective function after a certain number of iterations, together with a near-zero fitness variance, suggesting that the swarm reached a steady-state condition and that the optimization process had largely stabilized (see Figs. 3.8a, 3.8d, 3.8e, 3.8f, 3.8h, and 3.8i, as well as Figs. 3.9a, 3.9d, 3.9e, 3.9f, 3.9h, and 3.9i). This behavior is consistent with the swarm approaching a region of minimal objective function variation, indicative of near-convergent or quasi-stationary conditions.

In contrast, cases 4 and 5 show comparatively higher fitness variance values (see Figs. 3.8c and 3.8d, and Figs. 3.9c and 3.9d), suggesting that the optimization process had not yet stabilized within the assigned number of iterations. The persistent spread in objective function values indicates that the swarm was still exploring multiple regions of the design space, and that additional iterations could potentially have led to further improvement or stabilization of the solutions.

Similarly, in case 9, although the objective function appears to flatten after approximately 12 iterations (see Fig. 3.9g), the fitness variance remains on the order of 10. This residual variability suggests that the swarm had not fully converged to a single region of attraction, and that extended optimization could have further reduced the uncertainty

associated with the obtained solution.



(a) Case 3 (b) Case 4 (c) Case 5

(d) Case 6 (e) Case 7 (f) Case 8

(g) Case 9 (h) Case 10 (i) Case 11

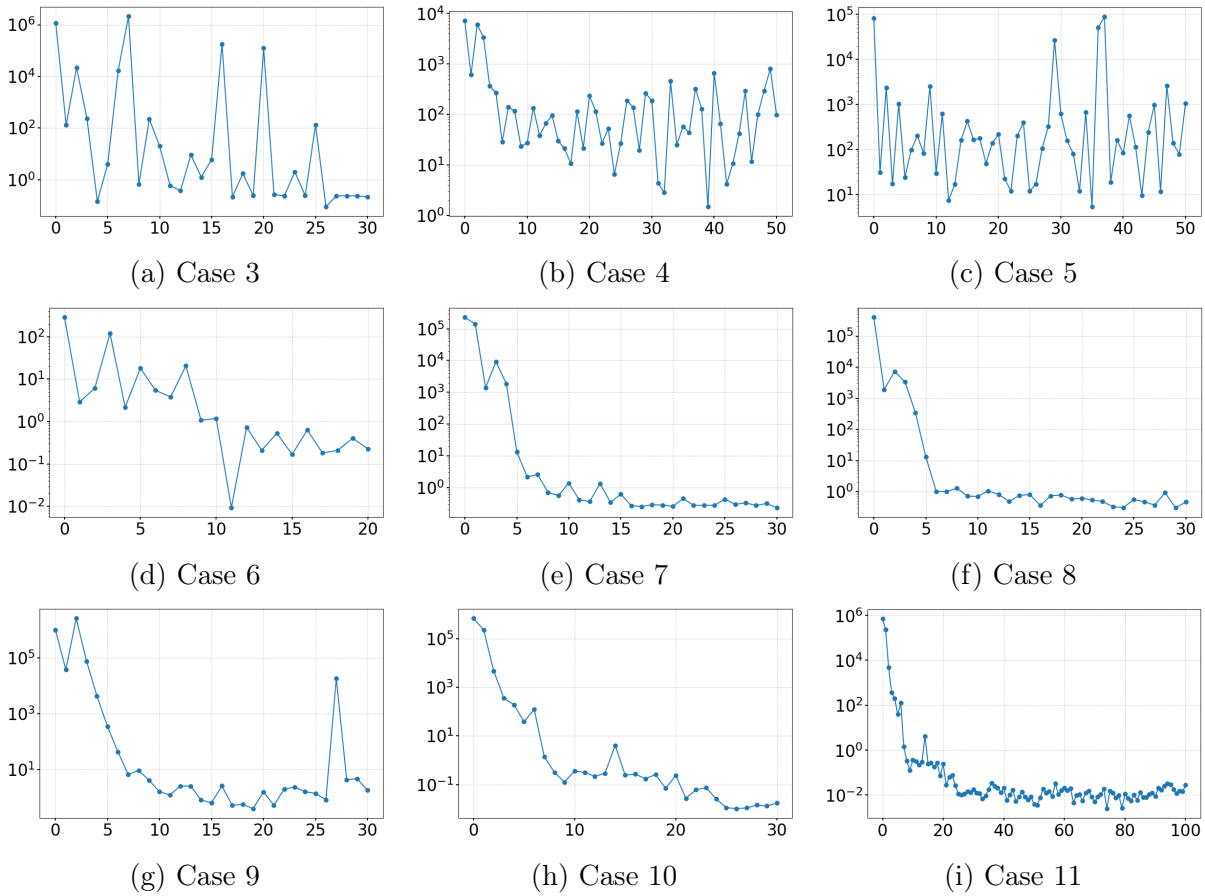
Figure 3.8: Normalized objective function *vs* iterations: 3.8a constant thickness; 3.8b-3.8c variable poloidal thickness; 3.8d-3.8e double wall shell with racetrack PRs; 3.8f-3.8g double wall shell with poloidal variable PRs; 3.8h-3.8i double wall shell with internal beams.

3.8 Conclusions

In this chapter, a comprehensive parametric modeling framework was developed to design the reinforcement system for a 10 T, 8.5 m bore Stellarator magnet. The modeling was carried out entirely within the ANSYS APDL environment, enabling efficient and automated analysis workflows.

To optimize the reinforcement configuration, the PSO algorithm was integrated with different parametric models.

Constraint-handling strategies were implemented to enforce physical and performance limitations, including bounds on DVs and stress constraints on the structure. A penalty function approach was adopted to steer the optimization away from infeasible regions while allowing a continuous search toward the optimal solution.



(a) Case 3 (b) Case 4 (c) Case 5

(d) Case 6 (e) Case 7 (f) Case 8

(g) Case 9 (h) Case 10 (i) Case 11

Figure 3.9: Normalized objective function variance across PSO swarm *vs* iterations: 3.8a constant thickness; 3.8b-3.8c variable poloidal thickness; 3.8d-3.8e double wall shell with racetrack PRs; 3.8f-3.8g double wall shell with poloidal variable PRs; 3.8h-3.8i double wall shell with internal beams.

The plot in Fig. 3.10 summarizes, for each case, the minimum structural mass required to satisfy the stress constraint (900 MPa) along with the corresponding maximum displacement. From the results, it is evident that the IS-OS-beams configuration does not meet the stress requirement (10^{th} and 11^{th} cases in Table 3.3), despite its mass being significantly higher than the other cases (exceeding 1800 tons per sector).

The most effective solution, which minimizes mass while still respecting the stress constraint, corresponds to a configuration where the IS thickness is constant ($t_{IS} = 543$ mm) and no additional supporting or reinforcing elements are included (3^{rd} case in Table 3.1). This design achieves a total mass of 1380 tons. However, such a large thickness poses major manufacturing challenges, as typical industrial capabilities are limited to plate thicknesses of 200–300 mm.

By constraining the IS thickness to a more realistic value of 300 mm, the optimal solution becomes the one in which the PR (and thus the OS) is allowed to vary poloidally. This configuration represents a balanced compromise between structural complexity, stress levels, maximum displacement, and overall mass, making it the most practical design choice

among those investigated (9^{th} case in Table 3.3).

An a posteriori assessment of the optimization process was also conducted using the fitness variance as an indicator of swarm stability. Although no explicit convergence criterion was applied, the combined analysis of fitness variance and objective function history allowed identification of cases that reached near-steady conditions and those still exhibiting residual variability. In particular, cases 4, 5, and 9 showed noticeable variance at the final iteration, suggesting that additional iterations would likely be required to achieve full stabilization of the swarm.

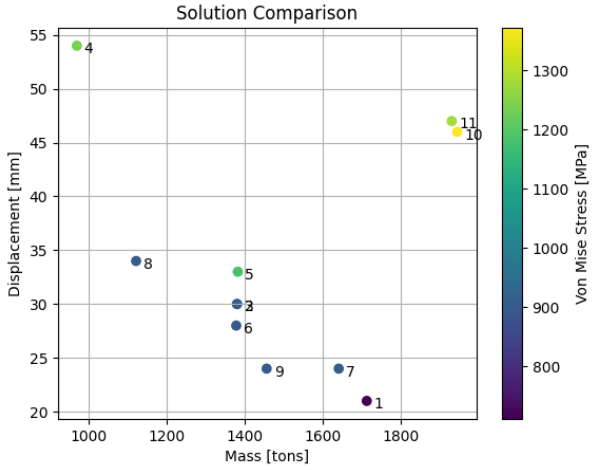
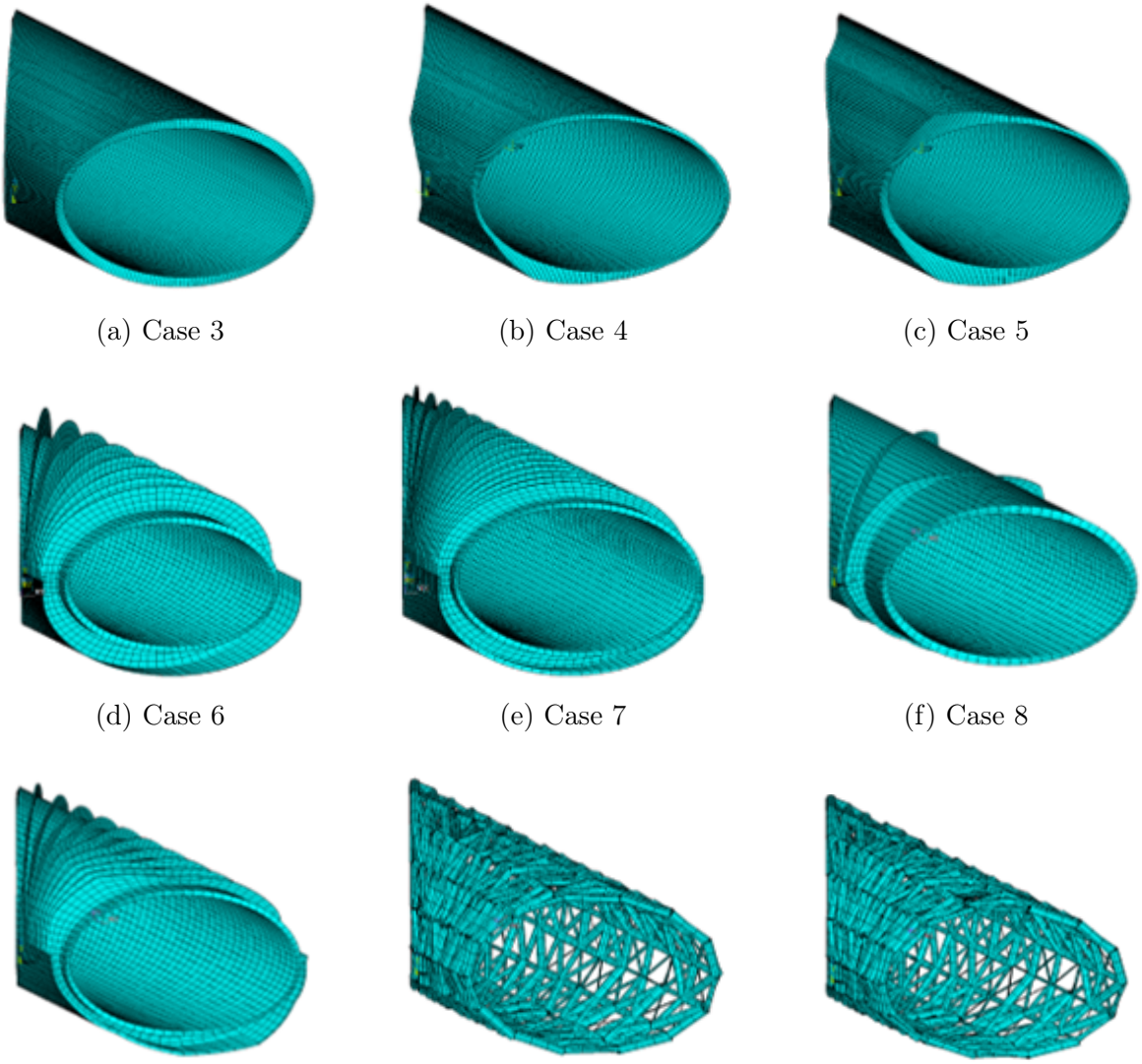


Figure 3.10: Results of parametric model in terms of mass (one sector), maximum global stress and maximum displacement.



(a) Case 3 (b) Case 4 (c) Case 5

(d) Case 6 (e) Case 7 (f) Case 8

(g) Case 9 (h) Case 10 (i) Case 11

Figure 3.11: Parametric modeling solutions: 3.11a constant thickness; 3.11b-3.11c variable poloidal thickness; 3.11d-3.11e double wall shell with racetrack PRs; 3.11f-3.11g double wall shell with poloidal variable PRs; 3.11h-3.11i double wall shell with internal beams.

4 Topology Optimization of the 10 T, 52 cm bore Stellarator magnet reinforcements

The development of next-generation stellarator experiments presents increasingly structural engineering challenges. Compact devices operating at high magnetic fields require highly optimized mechanical supports and reinforcements capable of withstanding extreme Electro-Magnetic (EM) loads without compromising manufacturability and system performance. Several fusion projects (both stellarators and tokamaks) have indeed highlighted that the peak stresses within coil support structures often approach the limits of structural materials [48, 49, 50].

Unlike the previous chapter, where the number of DVs was defined by the selected structural members, in this case the topology complexity was increased in order to move toward a more detailed and realistic experimental design concept. Such a design can serve as a smaller-scale prototype, with the potential to be scaled up for reactor-size magnets.

In this chapter, we present a comprehensive TO workflow, employing the PyMAPDL tool presented in sec. 1.6, for the design of magnet reinforcements conceived by RF. Unlike previous studies, which have focused on traditional HTS configurations, where reinforcement strategies follow relatively well-established structural design principles, our work addresses a novel challenge: the use of wide HTS tapes. This non-standard geometry significantly alters the mechanical load paths and creates new constraints on how reinforcement structures must be designed, assembled, and supported.

To tackle this, we developed a new methodology combining a novel implementation of simultaneous multi-directional extrusion constraints (see sec. 1.4.3)), iterative refinement of the design domain based on material efficiency, and a parametric study to translate TO outputs into manufacturable components. This approach enables the systematic exploration of lightweight, stiff, and robust reinforcement topologies capable of withstanding extreme EM forces. To the best of our knowledge, this represents one of the first applications of TO to the structural design of 10 T stellarator magnets. Beyond

identifying efficient reinforcement topologies, a key goal of this work is to ensure that the proposed structures can be translated into a manufacturable design. Particular attention is therefore given to incorporating manufacturing constraints and to developing an assembly strategy that bridges the gap between topology-optimized concepts and engineering solutions.

Here we apply this new TO method to magnetic confinement devices for the first time. As a first test application, we consider a hypothetical device of the same size and approximate geometry of an actual small test device under consideration at RF. However, in the present application of this chapter we consider a much higher field (10 T on the plasma axis) and therefore much higher forces than in the actual device. Also note that the coils lie on cylinders (see Fig. 4.1), whereas in future embodiments they might lie on slanted cylinders, which would require a revised TO. The device considered in this study can be regarded as a scaled version of the Wendelstein 7-X (W7-X) Stellarator in Germany [51], with the difference that the coil winding surface is piece-wise cylindrical rather than fully conformal to the plasma boundary. While this approach simplifies coil manufacturing, it introduces a non-uniform current distribution and, consequently, strong spatial variations in the EM forces that must be addressed by the structural reinforcements.

In the present study, the structural optimization problem can be formulated in principle as a mass minimization under mechanical performance constraints, namely maximum displacement below 1 mm, maximum magnet strain below 0.5%, and maximum von Mises stress below 800 MPa. However, in the current implementation we did not explicitly enforce these constraints in the topology optimization formulation. Instead, we adopted a stiffness maximization approach with varying volume fraction constraints (see sec. 1.4.1), which provides an equivalent pathway to identify efficient and lightweight structures (in second step of optimization we employed extrusion constraints, as shown in see sec. 1.4.3). The relationship between the imposed volume constraint and the resulting mechanical performance was subsequently analyzed through a sensitivity study, enabling us to identify configurations that satisfy the displacement, strain, and stress limits while maintaining minimum mass.

4.1 Magnet System Overview

A new class of stellarator magnets are being developed by Renaissance Fusion based on laser engraved wound HTS tapes. These magnets are designed to generate a target magnetic field of 10 T on the plasma axis, with a corresponding peak magnetic field of 15.8 T on the coil windings (Fig. 4.3). Such high field results in substantial EM forces,

placing significant mechanical demands on the supporting and reinforcement structures. To estimate the magnitude of these loads, the magnet can be approximated as a simplified helical solenoid. Using standard expression from electromagnetic theory, the magnetic pressure P exerted by a magnetic field can be calculated as:

$$P = \frac{B_0^2}{2\mu_0} \tag{4.1}$$

where B_0 is the magnetic field on the cylinder axis and μ_0 is the vacuum permeability ($\mu_0 = 4\pi 10^{-7}$ Tm/A). Assuming $B_0 = 10$ T at the plasma axis, the resulting uniform magnetic pressure on the windings is on the order of 40 MPa. This pressure generates substantial hoop and radial stresses that must be contained by structural supports to prevent failure or excessive deformation. The main parameters of the magnet are summarized in Table 4.1.

The mechanical challenge becomes even more complex when considering the overall magnet arrangement and load distribution. Five magnets will be positioned along a pentagonal path, as shown in Fig. 4.1, and will have a cylindrical shape, with an axial length of 50 cm and a diameter of 52 cm. The EM loads generated by the combination of the magnetic field **B** (see Fig. 4.3) and the density current **J** (see Fig. 4.2) produce an antisymmetric distribution, with an order of magnitude higher than 40 MPa, reaching locally peaks of several hundreds of MPa (see unwrapped cylindrical map of EM loads in Fig. 4.4); in practice, reproducing a stellarator magnetic field topology with a non conformal coil winding surface requires additional shaping of the current distribution (or equivalently the magnetic field). Radially, the centroids of each magnet are 65 cm and are equally spaced in the toroidal direction of 72°. Each magnet will be composed of wide HTS tapes with a thickness of 0.1 mm, forming a total thickness of 1.5 cm. The HTS laser engraved wide tapes will be wrapped on a former and we assumed the same thickness as the magnets for initial modeling. The former must carry and sustain the inward loads that come from the magnet, while the outer reinforcement structure must yield the loads pointing outward.

Table 4.1: Key parameters of the magnet.

Magnetic field on plasma axis, B_0	10 T
Peak magnetic field on coil windings	15.8 T
Magnet axial length	50 cm
Magnet diameter	52 cm
Radial centroid position	65 cm
Toroidal spacing	72°
HTS tape thickness	0.1 mm
Total winding thickness	1.5 cm
Number of magnets	5

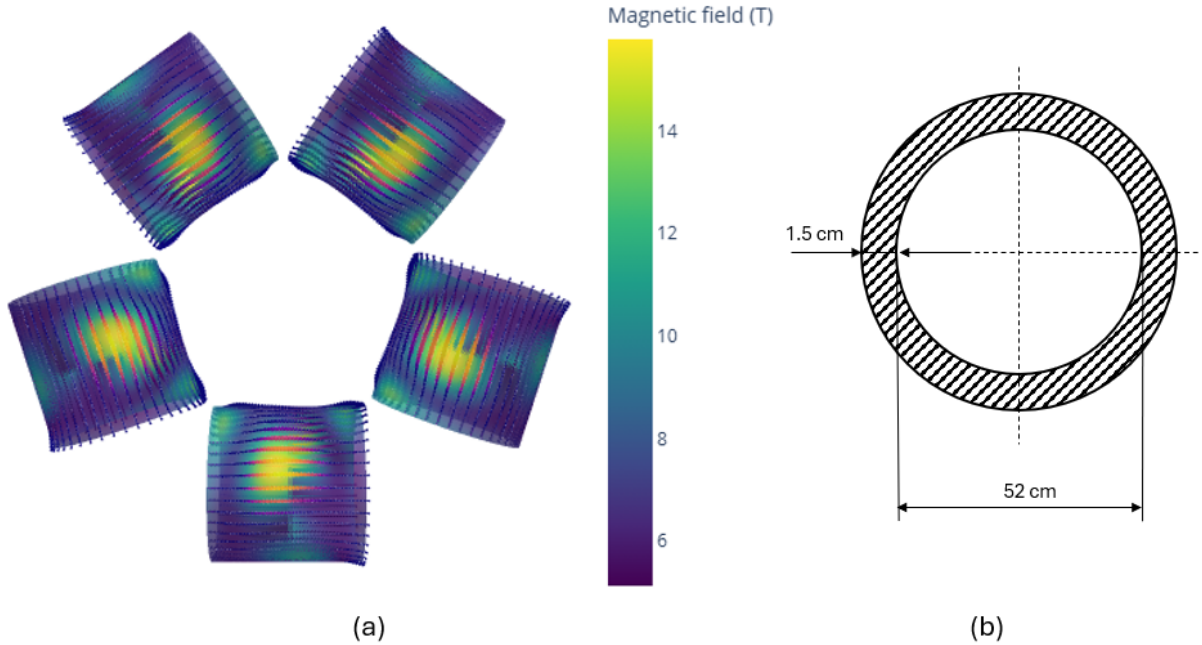


Figure 4.1: (a) Top view of magnet arrangements; (b) Magnet cross-section (not drawn to scale).

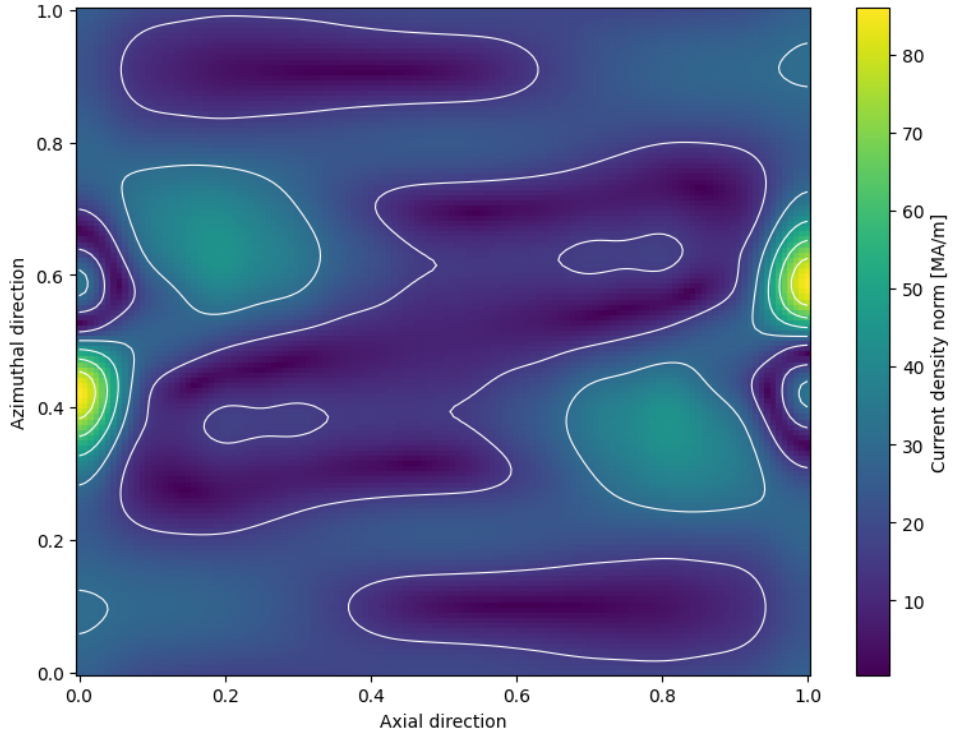


Figure 4.2: Two-dimensional distribution of the density current magnitude [MA/m], with axes normalized w.r.t. the magnet's axial and poloidal directions.

4.2 FE model of initial non-optimized structure

The structural Finite Element (FE) model consists of two concentric cylinders (magnet and former) which represent the non-design domain Ω_{ndd} surrounded by a larger domain

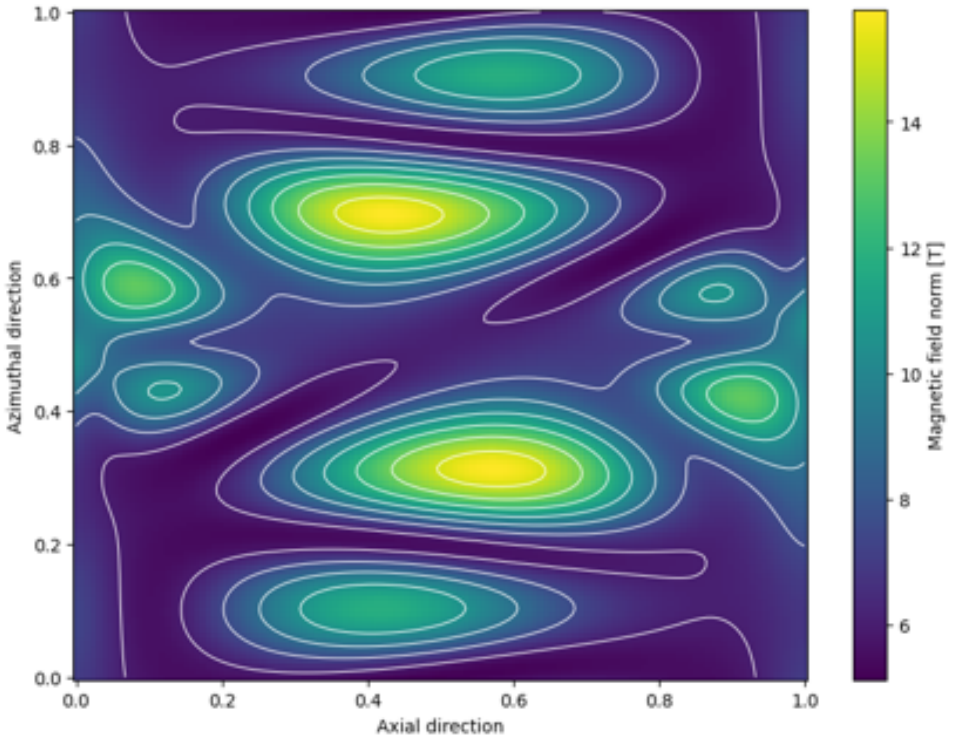


Figure 4.3: Two-dimensional distribution of the magnetic field magnitude [T], with axes normalized w.r.t. the magnet's axial and poloidal directions.

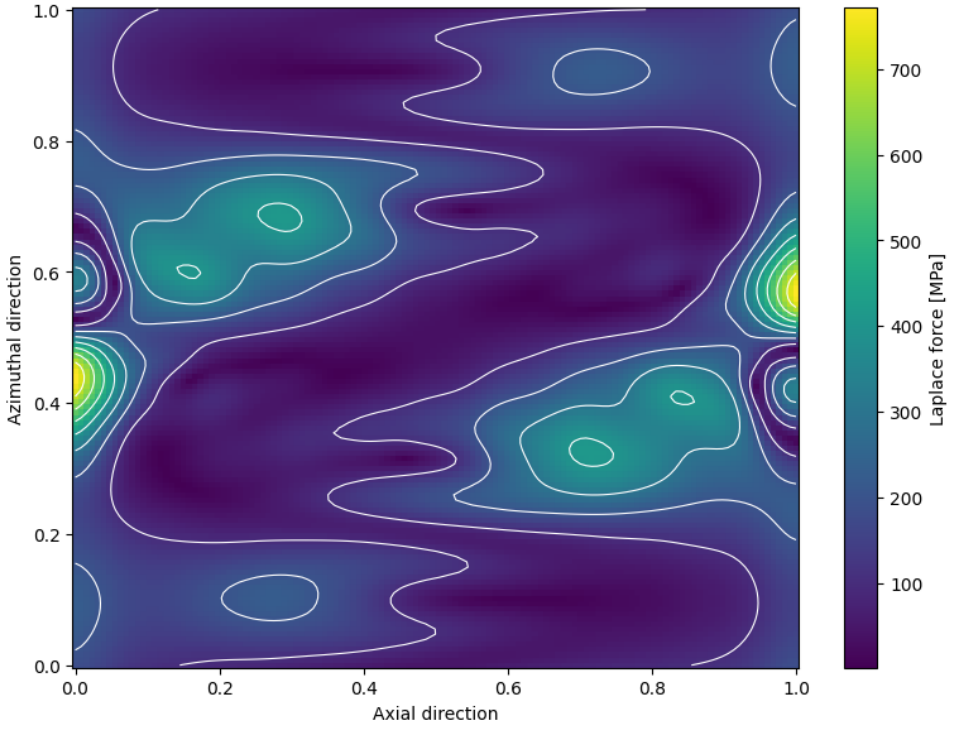


Figure 4.4: Two-dimensional distribution of the Laplace force magnitude [MPa], with axes normalized w.r.t. the magnet's axial and poloidal directions.

that defines the so called design domain region Ω_{dd} where the material must be removed, as shown in Fig. 4.5. The choice of this initial geometry is based on rational considerations. First, a sufficiently large design domain ensures that mechanical constraints such as maximum stress, displacement and strain remain within acceptable limits, even before optimization, thereby guaranteeing the feasibility of the starting model (providing also a larger margin when removing unnecessary material). Second, a generous initial volume provides the optimizer with enough freedom to systematically remove material, enabling the identification of efficient reinforcement layouts that would be difficult to obtain with a smaller design domain. Finally, the use of concentric domains reflects a practical assumption: in such application where the magnet behave like a pressure vessel, structural reinforcements are expected to surround the magnet itself, making this configuration a natural starting point.

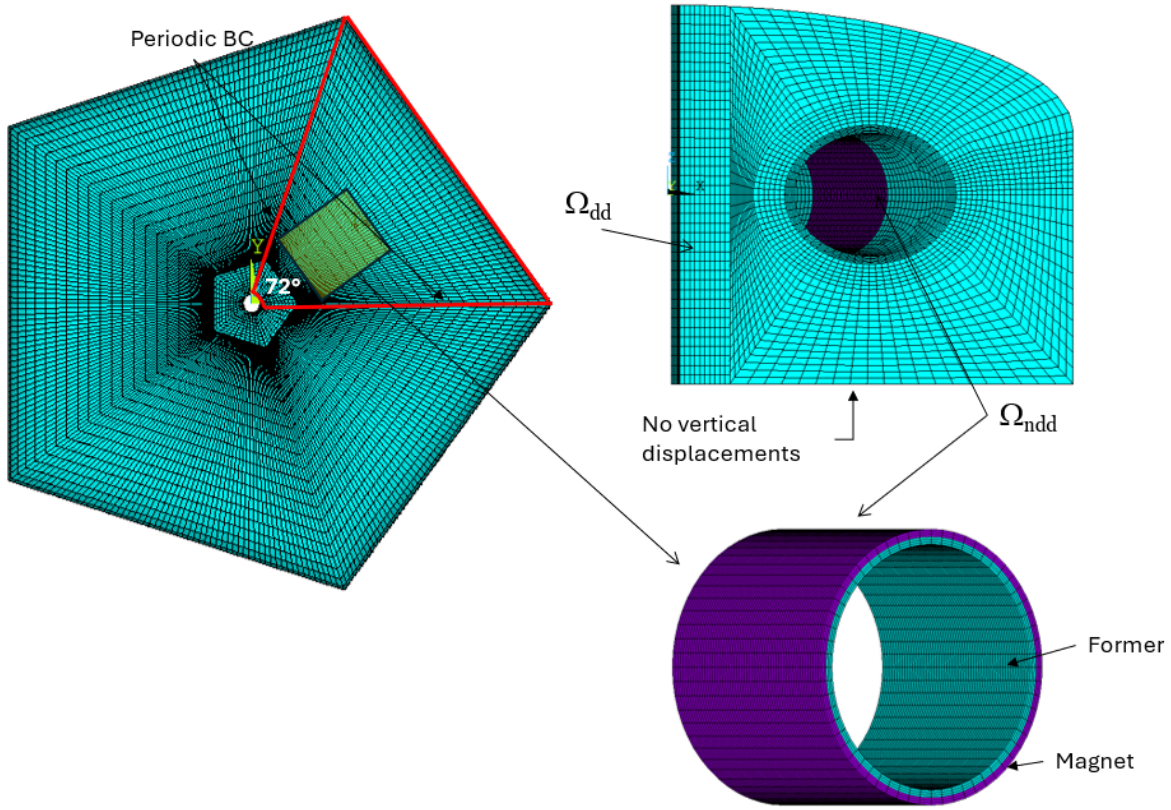


Figure 4.5: Structural FE model of magnet (Ω_{ndd}), former (Ω_{ndd}) and design domain (Ω_{dd}).

The initial mass considering both Ω_{ndd} and Ω_{dd} is about 11.3 tons. Both the former and the design domain are assumed to be made of SS316LN ($E = 200$ GPa, $\nu = 0.3$), a material widely adopted in cryogenic structural applications due to its favorable combination of mechanical performance and manufacturability. This steel is widely used in large-scale fusion systems such as ITER. At cryogenic temperatures, which match the operating range of the structure, this material exhibits a yield strength of around 1000

MPa at 4 K [47], making it particularly suitable for high-field, mechanically demanding environments such as this one. Although these limits were not explicitly included in the optimization formulation, the solutions presented in this work were selected to satisfy a conservative global stress limit of 800 MPa and a maximum magnet displacement of 1 mm. The effect of such displacements on magnetic field errors will be investigated in future studies.

The HTS magnets are composed of several layers of different materials, which are commonly a Hastelloy C-276 substrate (65- 75% of total HTS thickness), a buffer layer (2-5%), a REBCO superconducting layer (1- 2%), a stabilizer layer made of copper (15-25%) and an optional encapsulation layer (0-10%). Since a final HTS composition has not yet been specified, this multilayer configuration is based on representative architectures commonly found in existing commercial HTS tapes [52]. Because of this multilayer structure, the mechanical behavior of HTS materials is highly anisotropic, making direct modeling challenging, specifically with the various geometric scales. To address this, a homogenization technique was applied, allowing the magnet to be modeled as a transversely isotropic material [53, 54, 55]. In this model, mechanical properties are assumed to be constant in the plane of the HTS tape and variable in the out-of-plane (radial) direction. The following material parameters were defined for the HTS: radial (E_r), tangential (E_θ) and axial (E_z) Young's moduli; in-plane ($G_{r\theta}$) and out-of-plane ($G_{\theta z}$) shear moduli, in-plane ($\nu_{r\theta}$) and out-of-plane ($\nu_{\theta z}$) Poisson's ratios (see Table 4.2 for HTS equivalent mechanical properties). Additional details on how the homogenization technique is applied can be found in appendix .1.

Although the magnets will operate at cryogenic temperatures around 20 K, thermal contraction effects and thermomechanical loads have not been included in the present work. This assumption is justified by the focus of the study on EM load-bearing optimization during the early conceptual design phase, where stiffness and material distribution are the primary concerns. Moreover, in high-field fusion magnets, EM loads generally dominate the mechanical response.

Table 4.2: Equivalent mechanical properties of homogenized HTS material.

E_r [GPa]	E_θ [GPa]	E_z [GPa]	$G_{r\theta}$ [GPa]	$G_{\theta z}$ [GPa]	$\nu_{r\theta}$	$\nu_{\theta z}$
115	141	141	44	54	0.3	0.3

Due to the periodic symmetry of the EM loads produced by the magnets, the resulting forces generated by each coil are directed radially inward toward the central axis and repeat every 72° (see Fig. 4.1). This periodicity allows structural analysis and TO to be performed in a single representative sector, significantly reducing computational cost without sacrificing accuracy. The EM loads were mapped onto the inner magnet surface using ANSYS Workbench's interpolation tools. The components of the original

loads for the first magnet are F_x =-4.96 MN and F_y = -3.61 MN, respectively. Due to the mentioned periodicity of the loads, the total net force and resulting moment, when considering all modules, are zero, ensuring global static equilibrium.

The interaction between the former and the magnet, as well as between the magnet and the design domain, was initially modeled using bonded contacts. This simplification is appropriate for the early conceptual design stage, where the focus is on exploring optimal material distribution for structural stiffness and mass efficiency. Although bonded contacts may slightly overestimate load transfer efficiency and potentially increase local stress concentrations in the magnet, they allow for a more stable and computationally efficient simulation. Frictional contact, while more realistic for detailed mechanical behavior, was intentionally not implemented due to its high computational cost and slow convergence behavior, which can be prohibitive in iterative TO workflows. Regarding boundary conditions, the periodic nature of the geometry was enforced by applying periodic symmetry on the two lateral faces of the 72° sector, and constraint to prevent the vertical displacement were applied at the base of the design domain (see Fig. 4.5).

The displacement and stress fields of the initial non-optimized structure is shown in Fig. 4.6 and Fig. 4.7. The maximum displacement, that is lower then 1 mm, occurs in the magnet, while the most stressed components is the former which exhibits a peak stress about 530 MPa; furthermore, most of the stress is observed around the magnet and on its front side, reflecting the directional nature of the applied EM loads. An important safety parameter for the magnet is the longitudinal strain (along the tape-wrapping direction), which must be kept below 0.4–0.5% [56] to prevent irreversible degradation of the HTS material's current-carrying capacity. Although tangential strain is typically used in the assessment of narrow HTS tapes, in this study we opted to evaluate the equivalent strain. This choice reflects the consideration that in wide HTS tapes, deformation components in all directions can contribute to mechanical degradation. Moreover, to the best of our knowledge, there is lack of data or established criteria available in the literature for strain limits in wide HTS. Therefore, the equivalent strain offers a conservative and holistic metric for evaluating the mechanical behavior of the superconducting winding. In this case, the maximum equivalent strain in the magnet is about 0.34% (see Fig. 4.8), while the maximum stress on the magnet is about 410 MPa (see Fig. 4.7).

4.3 Conceptual design of magnet's reinforcements

The TO process for the design of the magnet's reinforcements was conducted in an iterative and hierarchical manner to progressively refine the material layout within the design domain. In the initial stage, the entire design domain was explored using a classical stiff-

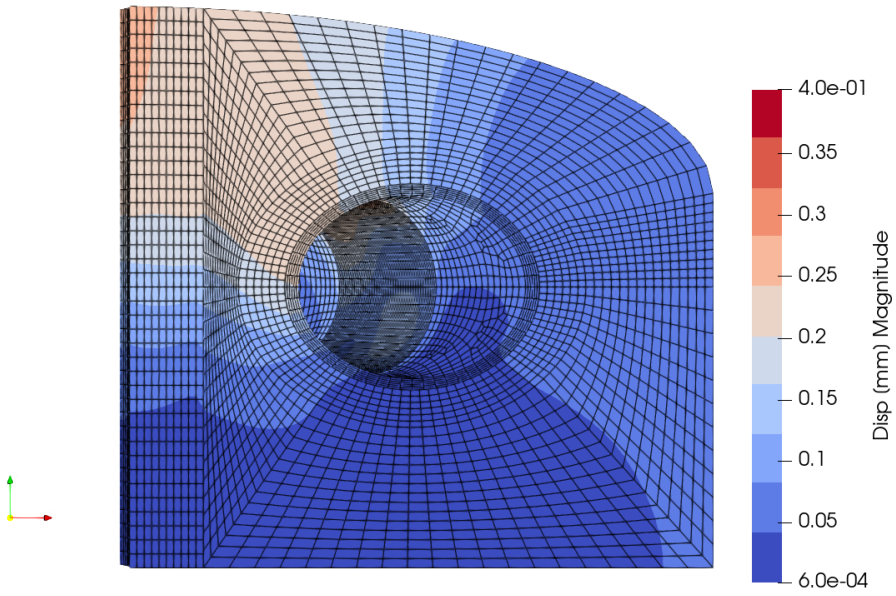


Figure 4.6: Displacement field [mm] of initial non-optimized structure.

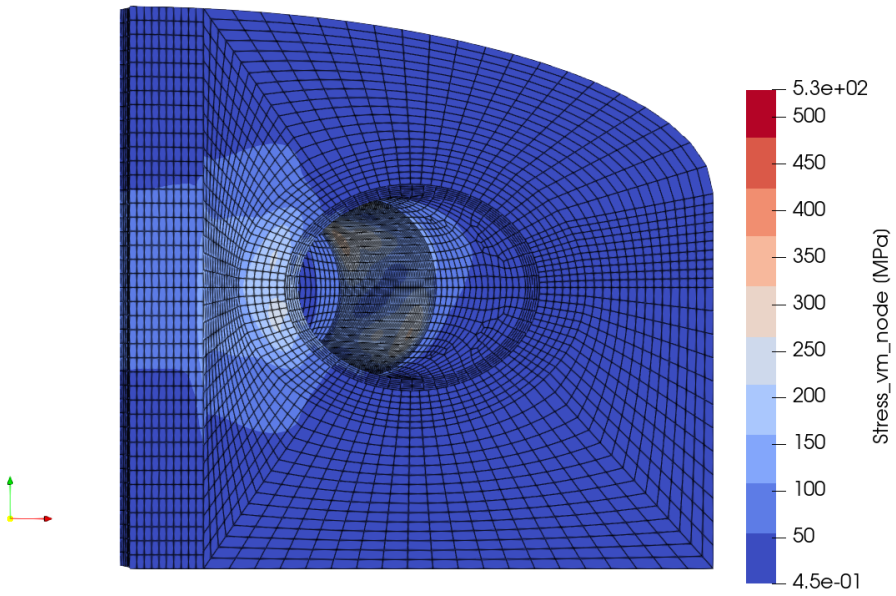


Figure 4.7: Stress field [MPa] of initial non-optimized structure.

ness minimization approach with only a volume constraint. This allowed the optimizer to identify the general load paths and primary structural features needed to withstand the EM loads. Based on the outcome of this first optimization, a more compact design domain was defined, excluding regions consistently identified as non-essential in the initial iteration. This updated domain was then used as the starting point for a second optimization run incorporating extrusion constraints, enabling the identification of key reinforcement patterns. Following this second iteration, a parametric model was developed by extracting and combining the key structural features observed in both TO outcomes. This model was used to refine and formalize the design, enabling controlled variation of geometrical

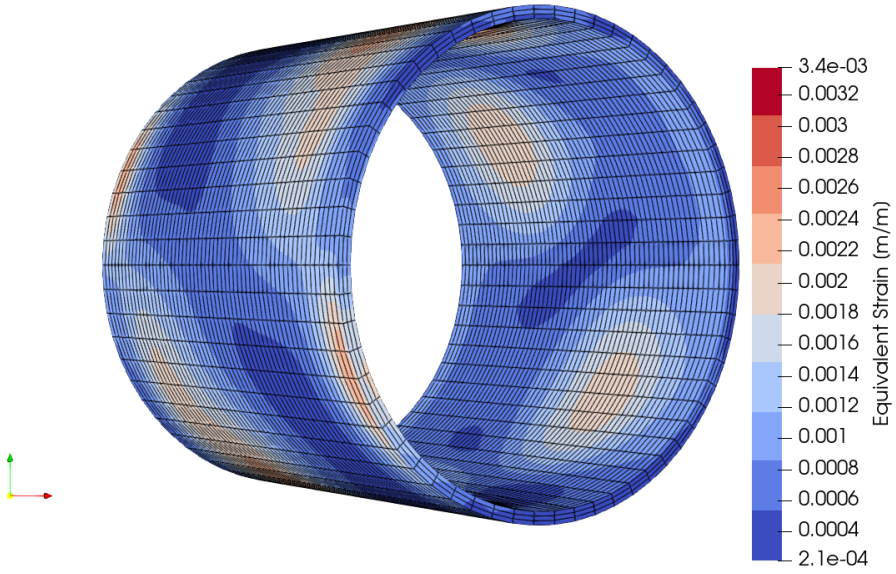


Figure 4.8: Equivalent strain field [m/m] of magnet (initial non-optimized structure).

parameters and further guiding the development of a manufacturable and mechanically robust solution.

4.3.1 Maximum Stiffness of initial model s.t. volume constraints

Three different volume fraction constraints were investigated: 30%, 20%, and 10%. These cases allowed for a comparative analysis of how material layout impacts the overall mechanical performance in terms of displacement, strain and stress. The outcomes are summarized in Table 4.3.

All resulting outcomes exhibited a similar qualitative behavior: material was mainly distributed around the magnet, forming a continuous reinforcement structure that resisted the EM forces. Notably, none of the configurations developed gravity support-like features, which is consistent with the nature of the net EM load being directed radially toward the machine center, and thus self-balanced in the full assembly.

Quantitatively, the solution at 30% volume fraction yielded a maximum displacement of 0.62 mm located on the magnet, a maximum equivalent strain of approximately 0.34% (also within the magnet), and a peak von Mises stress of 540 MPa in the former. The corresponding maximum stress within the magnet itself is 411 MPa. Reducing the volume fraction to 20% resulted in an increase in displacement to 0.81 mm and a slight increase in former stress to 566 MPa, with the magnet stress again reaching 422 MPa. The maximum equivalent strain remained essentially unchanged at 0.34%. Finally, under the most aggressive volume constraint of 10%, the maximum displacement increased to 1.5 mm, the equivalent strain in the magnet to 0.36%, and the peak stress in the former

reached 692 MPa. The corresponding magnet stress in this case is 463 MPa. Figures 4.9, 4.10, and 4.11 present the element density distributions obtained at the end of each TO run. In these figures, the non design regions are depicted in yellow.

Table 4.3: Mechanical response of initial model under varying volume constraints.

f [%]	m_{final} [t]	u_{mag} [mm]	$\varepsilon_{mag.}$ [%]	σ_{max} [MPa]
30	3.20	0.62	0.34	540
20	2.30	0.81	0.34	566
10	1.30	1.50	0.36	692

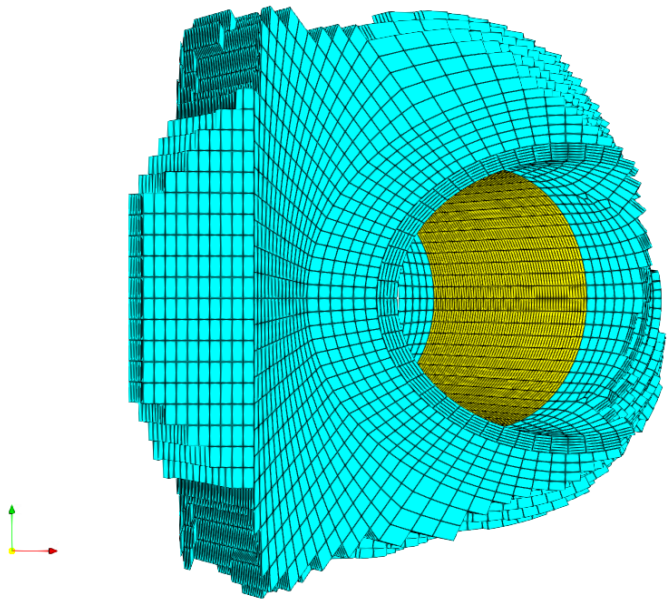


Figure 4.9: Element density plot of Initial Topology Optimized model with 30% of volume fraction ($m_{final} = 3.20$ t).

4.3.2 Maximum Stiffness of optimized model s.t. volume and extrusion constraints

Based on the obtained results in sec. 4.3.1, a new starting geometry (3.93 t) was defined to initiate the next stage of optimization (see Fig. 4.12). To simplify the shape while retaining key structural features, the material distribution from the first TO was projected onto the magnet's cross-sectional plane. This projection was then used to delineate the new design boundary. For clarity and manufacturability, the boundary was reconstructed using a B-spline representation controlled by a limited number of points, allowing the complexity of the geometry to be reduced.

Following the methodology described in sec. 4.3.2, several optimizations were performed by progressively modifying the extent of the non-design domain around the mag-

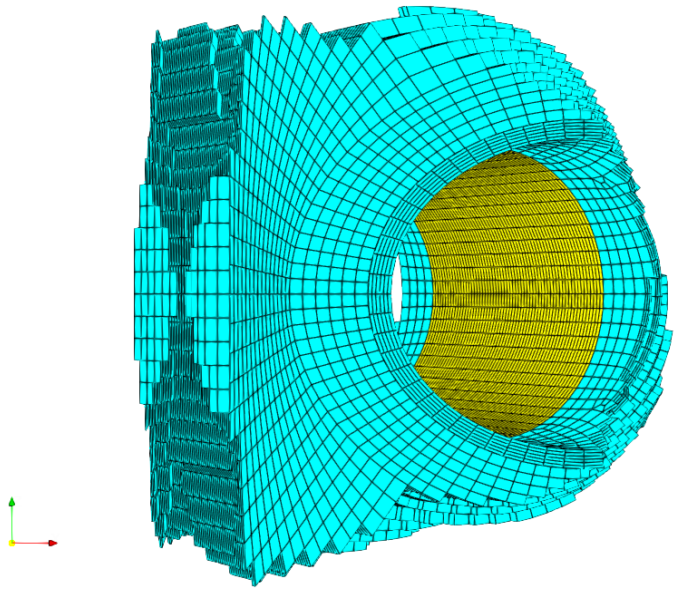


Figure 4.10: Element density plot of Initial Topology Optimized model with 20% of volume fraction ($m_{final} = 2.30$ t).

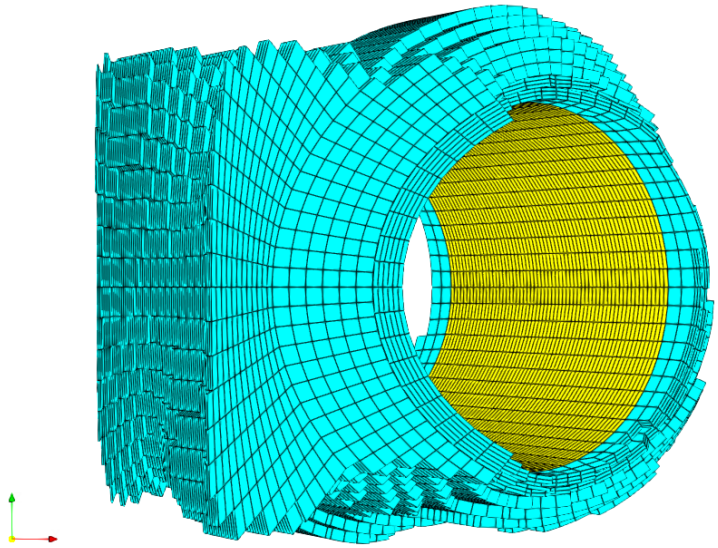


Figure 4.11: Element density plot of Initial Topology Optimized model with 10% of volume fraction ($m_{final} = 1.30$ t).

net and adjusting the volume fraction accordingly. This approach was adopted to assess how the imposed non-design domain influences the final material distribution and to better understand the impact of its size on the optimized structural layout. This adjustment ensured that all design cases resulted in comparable total masses, maintaining a consistent baseline for performance comparison. As in the previous analysis, each solution was

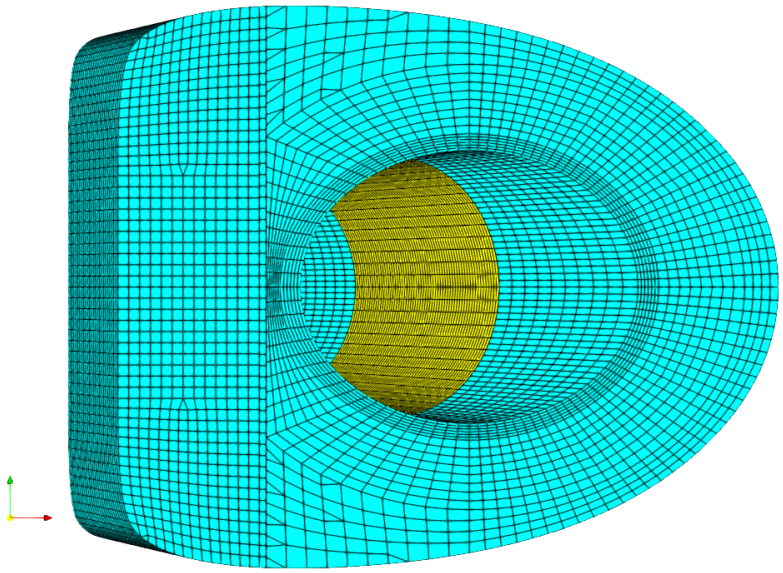


Figure 4.12: Reconstructed design domain for second-stage optimization.

evaluated in terms of material distribution, displacements, equivalent strains, and von Mises stresses, with particular attention paid to the structural response of the magnet.

The extrusion constraints were enforced simultaneously by partitioning the design domain into several distinct regions as shown in Fig. 4.13, each assigned to a specific extrusion direction. In more detail, the extrusion directions are radial (normal to the cylindrical magnet surface) for Region 1, and axial (along magnet axis) for Region 2-3-4. This method of enforcing multiple different extrusion directions enables more practical and manufacturable designs while preserving optimization flexibility and mechanical performance.

To constrain the model, one node of the design domain has been constrained in both toroidal and vertical directions, while bonded contacts have been again applied between each body in the FE model. The analyzed cases are shown in the first and second columns of Table 4.4. Here, n_{layers} is the number of layers and it is referred to the number of elements that surround the magnet and belong to the non design domain during TO process. The elements around the magnet are largely uniform in the direction normal to its respective outer surface, but tend to contract in Region 2. Approximately each layer of elements has a thickness of 1.6 cm.

The optimized material distributions are presented in Fig. 4.14. As shown in Fig. 4.14a–d, TO consistently promotes the formation of rib-like structures surrounding the magnet, providing enhanced stiffness along critical load paths. Additionally, all solutions consistently retain material at the front side of the magnet, as clearly illustrated in Fig. 4.14f. This recurring feature highlights its structural importance across different configurations

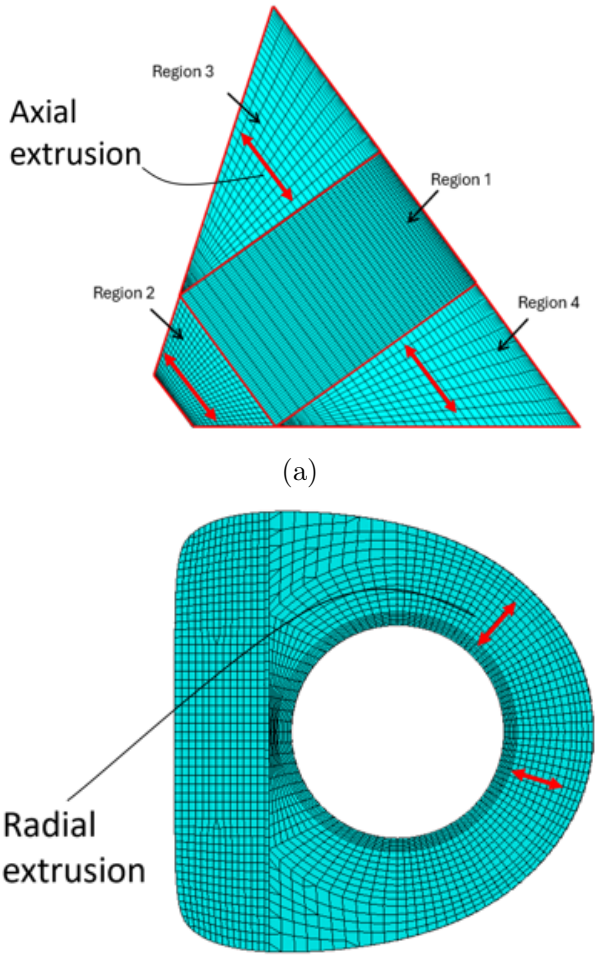


Figure 4.13: Extrusion regions.

and suggests the necessity of incorporating a central ring in the final design. It is also evident that the majority of material tends to concentrate in regions 1 and 2. Minor differences in mass (see Table 4.4) across analogous cases are due to the relatively large tolerance set for the termination criteria ($\tau = 0.1$). This choice enabled faster convergence of the optimization process without significantly affecting the quality or reliability of the results.

As the thickness of the material surrounding the magnet increases, both the maximum displacement within the magnet and the peak stress in the structure tend to decrease. In terms of equivalent strain, values remain largely consistent across most cases; however, a noticeable increase is observed in the configuration with three layers and the lowest volume fraction due its lower stiffness.

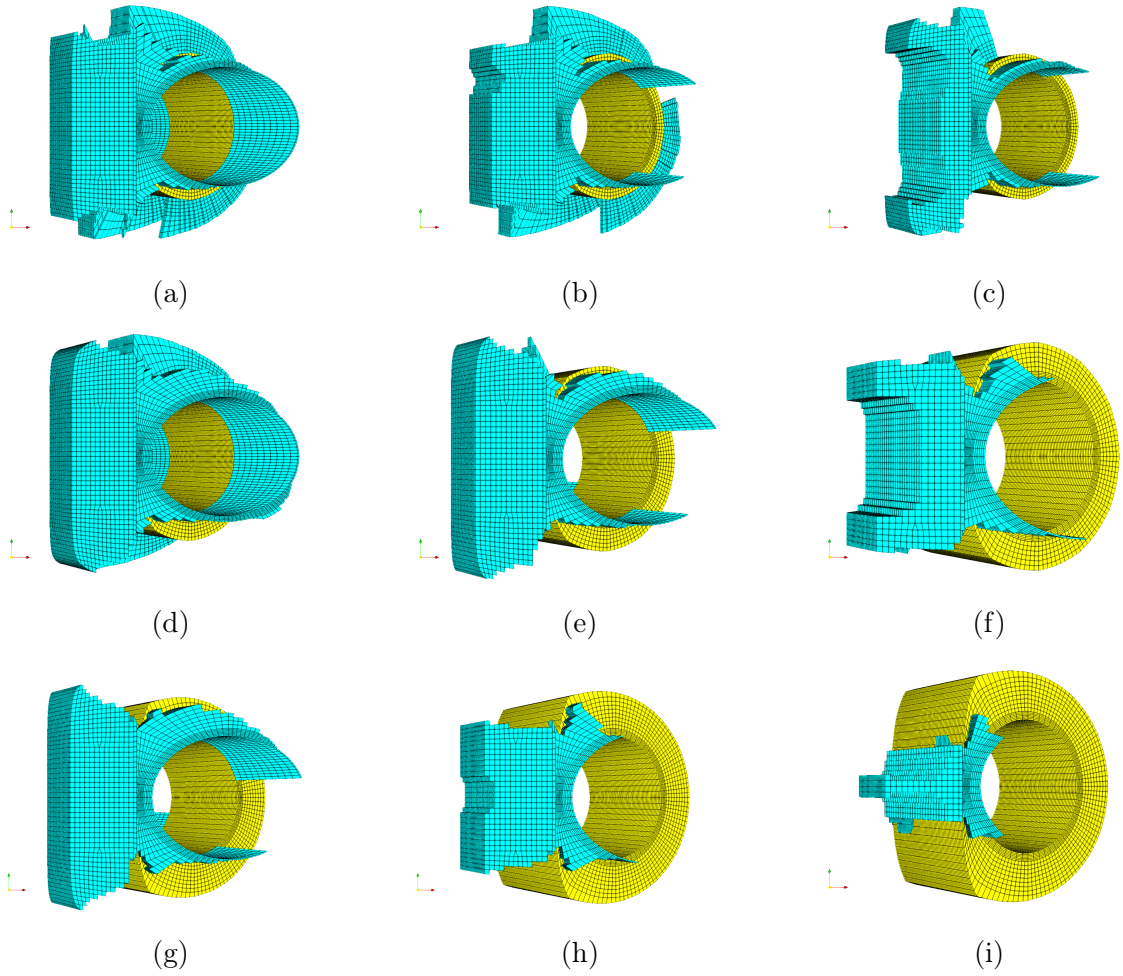


Figure 4.14: Topology optimization solutions with volume and extrusion constraints. Figures (a)–(c): 3 layers, volume fractions 50%, 37.5%, 25%; Figures (d)–(f): 6 layers, 45%, 31%, 17%; Figures (g)–(i): 9 layers, 36%, 21%, 5%.

n_{layers}	f [%]	m_{final} [t]	$u_{mag.}$ [mm]	$\varepsilon_{mag.}$ [%]	σ_{max} [MPa]
3	50	1.89	0.64	0.37	620
	37.5	1.57	1.30	0.36	720
	25	1.17	1.60	0.47	886
6	45	1.99	0.62	0.36	600
	31	1.58	0.83	0.37	710
	17	1.22	0.88	0.38	793
9	36	1.94	0.57	0.34	620
	21	1.62	0.60	0.35	670
	5	1.33	0.65	0.37	780

Table 4.4: Topology optimization results for different extrusion constraints. Here, n_{layers} denotes the number of extrusion layers and f the applied volume fraction. Reported values include final mass m_{final}, maximum magnet displacement $u_{mag.}$, equivalent strain $\varepsilon_{mag.}$, and maximum stress σ_{max}.

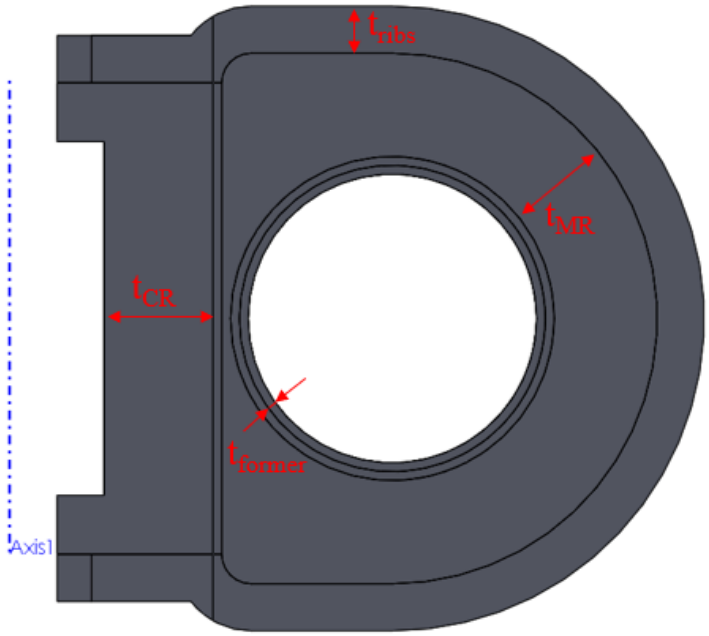


Figure 4.15: Lateral view of the parametric model for DoE. t_{MR}: Magnet Reinforcement thickness; t_{former}: former thickness; t_{CR}: Central Ring radial thickness; t_{PR}: Poloidal Ribs radial thickness.

4.3.3 Transitioning from TO to Parametric model

Here, the objective was to minimize the structural mass ensuring that the displacement of the magnet was below 1 mm, while its strain under 0.5%, and the maximum global stress lower then 800 MPa.

To transition from the TO results to a manufacturable structural concept, a parametric model was developed. The goal was to capture and combine the most relevant geometric features consistently observed across the optimized solutions, such as the presence of reinforcing ribs and the central ring (see Fig. 4.14).

The parametric model comprised of a Magnet Reinforcement (MR) with two Poloidal Ribs (PRs) at each end, and a Central c-shaped ring (CR), as shown in Fig. 4.15. Consistent with sec. 4.3.2, one node is constrained in both toroidal and vertical directions, and periodic symmetry were applied.

The key design parameters selected to define the optimal configuration include the Magnet Reinforcement thickness (t_{MR}), the former thickness (t_{former}), the radial thickness of the Central Ring (t_{CR}), and the radial thickness of the ribs (t_{ribs}), while the remaining geometrical parameters have been set according to the TO solutions (see Fig. 4.15 for selected parameters).

The impact of the selected parameters on the structural performance was assessed using ANSYS Design of Experiments (DoE), which enabled a systematic exploration of

the design space through a series of parametric simulations, facilitating the identification of optimal configurations. The results obtained are shown in the plots of Fig. 4.16 and Fig. 4.17.

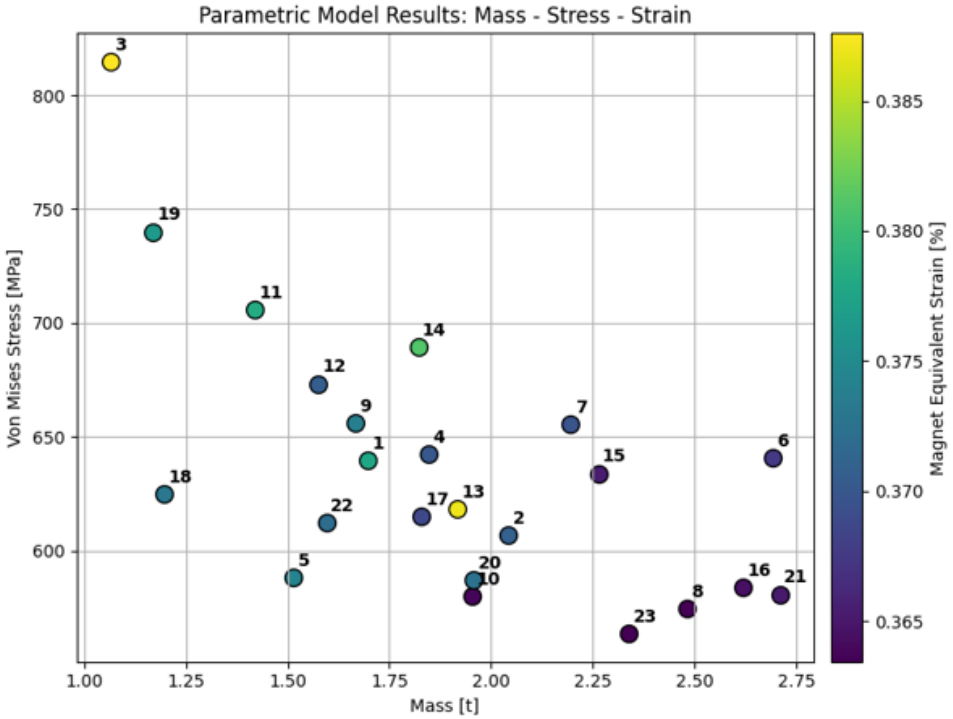


Figure 4.16: Results of parametric model in terms of mass (one sector), maximum global stress and maximum magnet strain.

Each solution in Fig. 4.16 and Fig. 4.17 is referred to a combination of t_{MR}, t_{former}, t_{CR} and t_{ribs}, obtained with the Latin Hypercube Method (LHM). Additional bound limits on the selected variables were included to promote a feasible design. Nearly all evaluated design points satisfied the imposed constraints, with the exception of solution 3, where both maximum displacement and stress slightly exceeded acceptable limits. Among all the evaluated design points, solutions 5 and 18 emerged as the most promising. While solution 18 exhibits marginally higher displacement (0.7 mm vs. 0.5 mm for solution 5) and stress (625 MPa vs. 588 MPa), its equivalent strain remains comparable to that of solution 5 (0.37%). Notably, solution 18 achieves this performance with a reduction of nearly 300 kg in total mass, making it an attractive lightweight alternative.

4.3.4 Proposed design of magnet's reinforcements

The selected design (solution 18) was refined to ensure well-defined, manufacturable structural features and to prevent any interference with adjacent sectors (see Fig. 4.18). Specifically, the PRs were positioned slightly inward from the ends to mitigate potential interference caused by manufacturing tolerances and deformation during operation. This

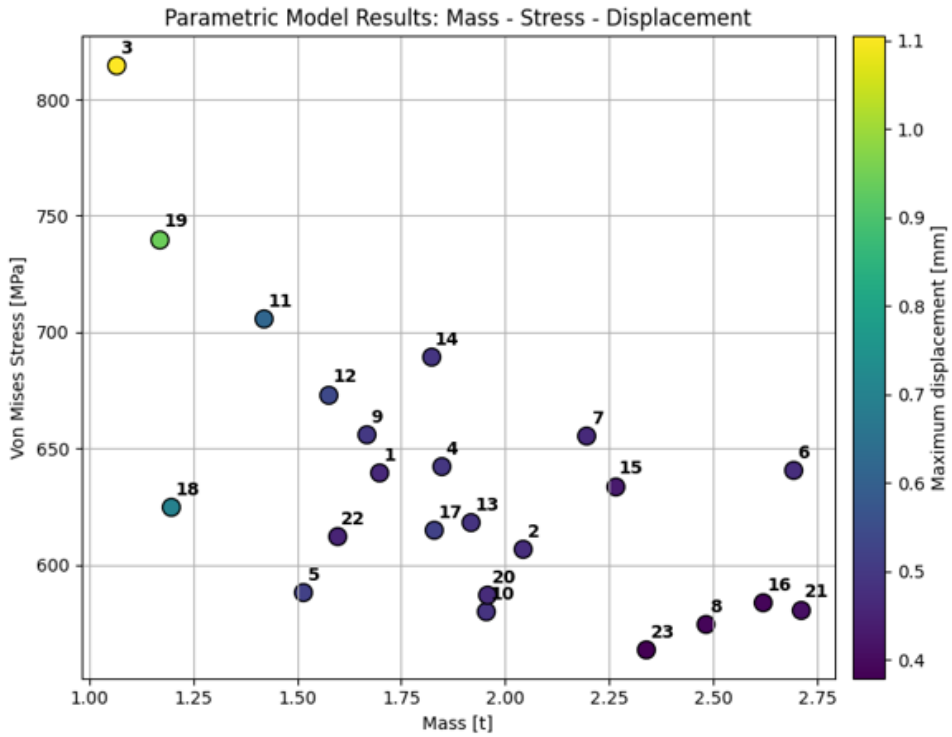


Figure 4.17: Results of parametric model in terms of mass (one sector), maximum global stress and maximum magnet displacement.

offset also facilitates assembly: the PRs are intended to be directly welded onto the outer MR surface, but a small clearance is necessary to slide them into place before welding. Therefore, a gap is intentionally provided at both ends of the MR to allow sufficient space for the weld seams.

A central hole was introduced in the CR to ensure tangential continuity, thereby eliminating sharp corners and reducing stress concentrations.

Additionally, two plates were designed to fully enclose the magnet and former, effectively preventing possible axial motion (along magnet axis). The former will be directly connected to these plates through bolted joints, ensuring secure and reliable attachment. A further structural element, referred to as the Outer Cylinder (OC), was introduced to fully enclose and protect the magnet during the fitting operation within the MR.

The thickness of the CR were reduced by increasing the MR thickness towards the machine center, allowing the possibility of assembling the two components using bolted connections. Moreover, the CR were furthermore divided in three main components, that are vertical plates which will be joined with the MR and two upper and bottom horizontal plates. The bottom plate, with a pentagonal shape, will serve as a base for the vertical plates; the two components can be jointed either with bolt connections or weldings. In the first case, the assembly is clearly more modular, however the bolts will carry most of the applied loads. In contrast, the welded connection offers better load distribution due

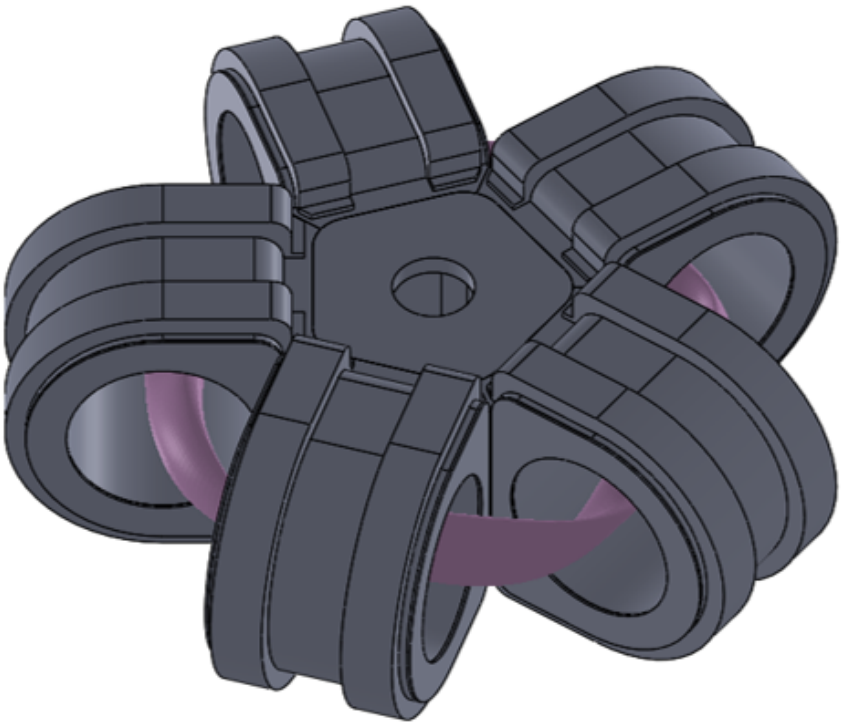


Figure 4.18: Representation of the conceptual design of the magnet's reinforcements.

to greater continuity and more effective load sharing across the entire joint. However, if the weld joints are large and deep, heat treatment becomes necessary to relieve residual stresses and ensure proper mechanical performance. If the HTS cannot tolerate elevated temperatures, which may be the case, the welding process would need to be carried out without the magnets in place. This condition introduces a significant constraint on the overall manufacturability of the structure.

Lastly, the double symmetry of the MR allows for efficient production via casting. The inner surface, which interfaces with the OC, can be further refined to achieve the required surface roughness.

4.3.5 Assembling procedure

In this section, we outline a proposed assembly procedure to ensure that all components can be integrated seamlessly and achieve the desired structural integrity under operational conditions. An exploded view of the magnet reinforcements components is depicted in Fig. 4.19.

The first step consists of winding the wide, engraved HTS tape around a cylindrical former with suitable winding tension. This operation results in the creation of the magnet–inner former unit. Once the winding is complete, the entire assembly is cooled down to facilitate the next step, which involves inserting this unit into the OC. This

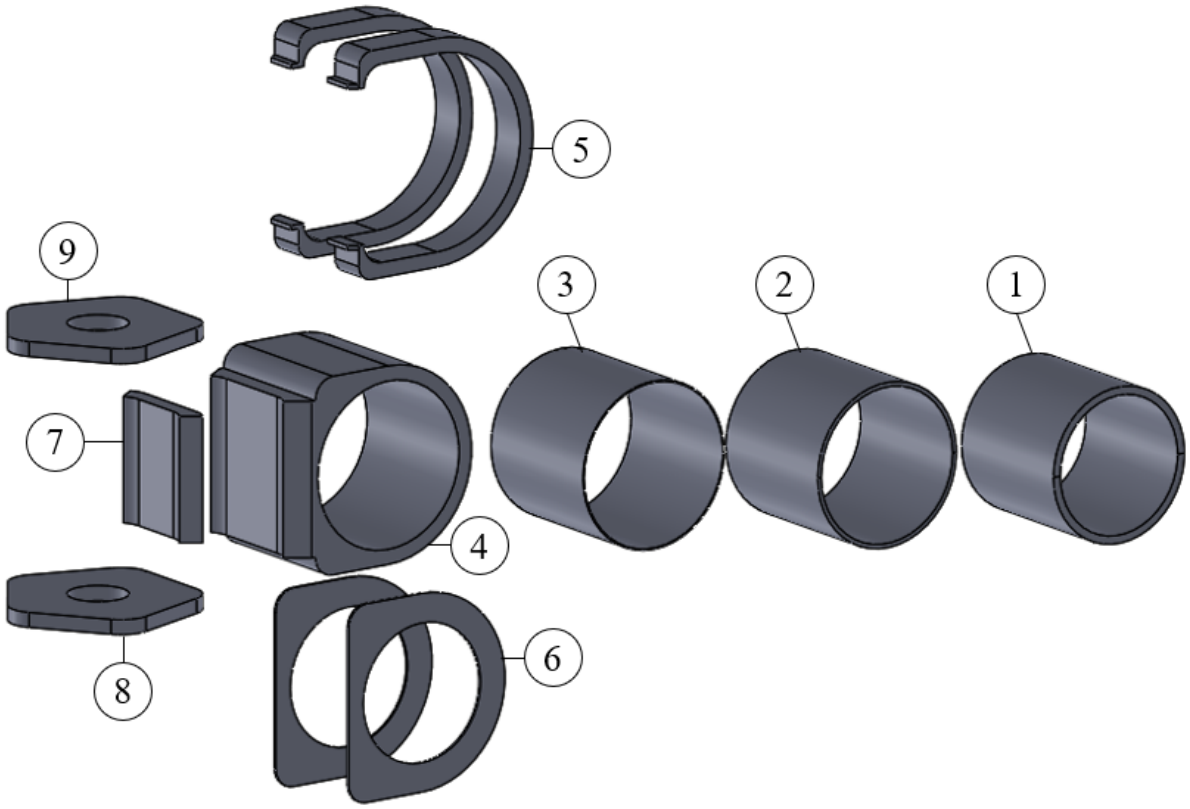


Figure 4.19: Exploded view: 1) former, 2) magnet, 3) OC, 4) MR, 5) PRs, 6) former caps, 7) CR vertical plate, 8) CR bottom plate, 9) CR upper plate.

step requires a slight interference fit to ensure a secure and rigid connection between the components.

After assembling the magnet, inner former, and OC into a unified structure, this composite unit is inserted into the MR. This second insertion can be achieved through either a classical interference fit or a tapered fit. In the case of a conical (tapered) coupling, cooling of the internal components can often be avoided, as the angled surfaces allow insertion by applying an axial force, which generates the required contact pressure between components. The efficiency of this method, however, depends heavily on the taper angle, interference magnitude, and surface roughness (an analytical method to quickly estimate the the sliding force and the pressure contact is described in appendix .3). Once inserted, two additional plates are attached to the lateral faces of the module. These plates serve to fully enclose the magnet—thereby minimizing possible axial movement—and mechanically connect the inner former to the MR using bolted joints. The MR also extends toward the machine center to provide sufficient material thickness to host radial bolts used for connecting to the CR.

Following this step, the CR assembly can be performed independently. The CR consists of a bottom pentagonal plate, five vertical plates, and a top plate matching the bottom geometry. These components are joined using either bolted or welded connec-

tions, depending on the desired mechanical performance and modularity.

Then, the MR and CR assemblies are connected using radial bolts inserted through the inner sides of the CR's vertical plates. Finally the upper plate is bolted on CR's vertical plates.

Future design could include a larger circular hole for the upper plate, which could be welded with the vertical plates for better structural connection. This should also provide sufficient space for inserting the radial bolts. Another option that could increase the CR stiffness, consists of directly casting all components (vertical, bottom and upper plates), eliminating the need for precise welded connections between the plates.

Even if the primary considered approach to assemble the magnet within the MR is a fit interference, an alternative may involve splitting the MR into two halves and joining them using bolted connections. A high-level calculation of the segmentation strategy and the corresponding number of bolts required is presented in appendix .2.

4.3.6 FEA of conceptual design

To evaluate the effects of the design changes described in sec. 4.3.4, a FEA was carried out using boundary conditions similar to those detailed in sec. 4.3.2, with the exception of contact definitions used to model the interactions between the OC, the magnet, and the MR. Specifically, a bonded contact was assumed between the magnet and the former, reflecting the presence of tape that will be secured to the former. In contrast, frictional contacts were defined at the interfaces between the OC and the magnet, as well as between the OC and the MR, using a friction coefficient of $\mu_s = 0.15$. A preliminary interference value of $i = 0.05$ mm was assumed in this study for the fit interfaces; however, dedicated analyses are required to determine the optimal interference values to guarantee consistent contact, particularly during cool-down and to avoid separation between components under operational conditions.

The magnet undergoes a maximum displacement of 1.1 mm (see Fig. 4.20), with a peak strain of 0.50% and a corresponding maximum von Mises stress of 648 MPa (see Fig. 4.21). The highest stress within the entire assembly occurs in the lateral plates, where the stress reaches up to 870 MPa.

Although the peak stress slightly exceeds the value obtained with the parametric model, it remains below the acceptable limit of 1000 MPa. The magnet displacement, which is nearly two times greater, could be reduced by increasing either the PRs dimensions or the MR thickness. Finally, the magnet strain can be further minimized by increasing the interference fit between components. These observations suggest that a parametric sweep, which can be readily performed in ANSYS, would be an effective way

to systematically evaluate the influence of these variables. Such an approach can be viewed as a brute force optimization method, yet it remains perfectly suitable given the limited number of parameters involved.

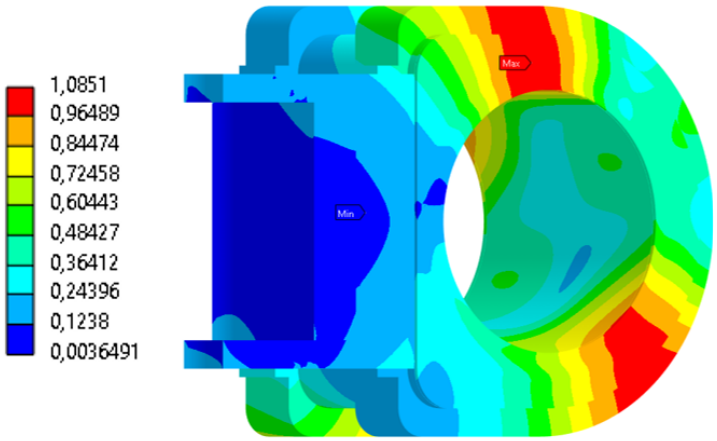


Figure 4.20: Displacement field of proposed design [mm].

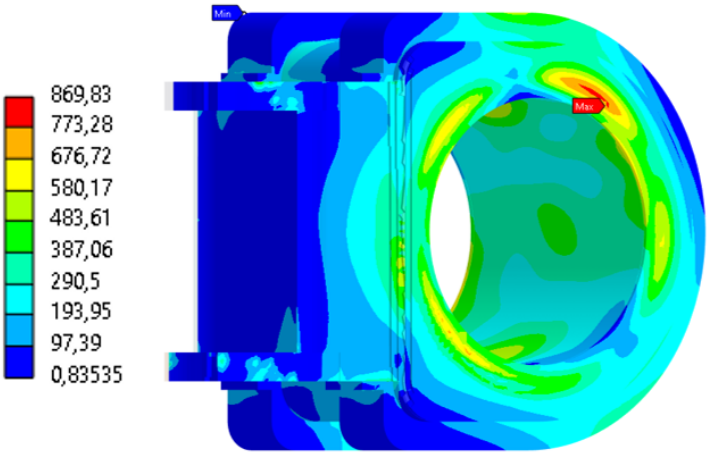


Figure 4.21: Stress field of proposed design [MPa].

4.4 Conclusions

In this chapter, a structural optimization framework was developed for the conceptual design of reinforcement structures for a 10 T on axis, 52 cm bore stellarator magnet. The design process began with an initial mass minimization strategy, which led to a significant reduction in the design space, from 11.3 t to 3.93 t. A second round of TO, incorporating multiple extrusion constraints, further reduced the mass to 1.65 t and revealed key structural features such as a C-shaped Central Ring (CR), Poloidal Ribs (PRs), and a Magnet Reinforcement (MR). A parametric model was then created based on the second

optimization run to explore design variants and refine key parameters, ultimately achieving a final mass of 1.20 t. To accommodate manufacturing and assembly requirements, the design was slightly adjusted without compromising its structural integrity. The final proposed configuration experiences a maximum displacement of 1.1 mm, a peak strain of 0.50%, and a von Mises stress of 648 MPa in the magnet. The highest stress in the assembly (870 MPa) is reached in the lateral plates. Compared to the initial large-scale design, this approach resulted in a nine-fold reduction in mass, while maintaining acceptable mechanical performance.

Future developments should focus on the detailed design of sub-components through high-fidelity FEA, including the modeling of bolts, and welds. Additional efforts should be directed toward optimizing interference fits to achieve the desired magnet pre-compression. Another promising direction is the exploration of higher-dimensionality TO, for example using beam or truss representations; this could enable to reach lighter structures, while also allowing for asymmetric configurations that may exhibit superior structural performance. Furthermore, robustness analyses should be conducted to assess the sensitivity of the optimized reinforcement structures to variations in the magnetic field map, ensuring that small deviations do not compromise the overall mechanical performance. Finally, a thermo-mechanical analysis under realistic operating conditions is essential to evaluate potential contact losses or mechanical degradation during cool-down phases and operational conditions.

5 Conclusions and Future Works

Conclusions

This thesis focused on the development of optimization tools for the structural design of Renaissance Fusion stellarator magnet reinforcements. Two complementary approaches were explored: parametric sizing–shape optimization for a larger 10 T, 8.5 m bore magnet, and Topology Optimization (TO) for the conceptual design of a 10 T, 52 cm bore magnet. Together, these studies demonstrated how advanced optimization strategies can guide both early stage design exploration and later stage refinement under realistic engineering constraints.

In chapter 3, several parametric models were developed in ANSYS APDL and design exploration were addressed with a Particle Swarm Optimization (PSO) algorithm. Constraint-handling strategies were also implemented to enforce both design variable bounds and structural stress limits (as penalization on the objective function), ensuring realistic optimization under engineering conditions. The results indicated that solution consisting of Inner Shell (IS),Outer Shell (OS) and beam configurations are excessively heavy, with masses exceeding 1800 t per sector, and fail to satisfy the required stress constraints. The most efficient solution was obtained with a thick IS ($t_{IS} = 543$ mm, 1380 t), although such a configuration is impractical from a manufacturing standpoint. Considering a more realistic thickness limit of 300 mm, designs incorporating variable Poloidal Ribs (PRs) and OS reinforcements provided the best balance among structural complexity, stress distribution, displacement, and manufacturability. Moreover, it is worth noting that few of the parametric optimizations did not fully reach convergence, as the number of iterations was intentionally limited to balance computational effort and turnaround time. Consequently, a potential direction for future work would be to rerun these unconverged cases with a larger number of iterations, enabling a more consistent and comprehensive comparison among the different parametric model outcomes.

In chapter 4, the conceptual design via TO study identified key structural features that include a central ring, poloidal ribs, and magnet reinforcements, that significantly reduced mass while maintaining structural integrity. The developed framework lowered

the reinforcement mass from 11.3 t to 1.2 t (a reduction of nearly 90%), while keeping displacements (1.1 mm), strain (0.5%), and stress levels (648 MPa in the magnet and 870 MPa in the lateral plates) within acceptable limits. These results demonstrated that early-stage TO can effectively guide conceptual design by revealing optimal load paths and essential reinforcement regions.

Overall, the thesis demonstrated how optimization-based methodologies, when coupled with Finite Element Analysis and practical engineering constraints, can significantly reduce structural mass, identify efficient reinforcement layouts, and provide a foundation for manufacturable stellarator magnet reinforcement designs.

Future Works

The research carried out in this thesis demonstrated the potential of optimization-driven methods for the design of magnet reinforcement systems. Nevertheless, several aspects deserve further development in order to increase the maturity and applicability of the proposed approaches:

1. **Stress-Constrained implementation in Topology Optimization tool (chapter 4):** in this work, the topological optimization approach was performed without an explicit stress constraint. A natural extension is the integration of stress-based formulations, allowing the optimization process to directly account for mechanical strength limitations. Particular emphasis should be pointed to fully replicate the low level FE matrices of ANSYS software. This would lead to less design sensitivity, so does reducing the computational time to achieve a conceptual design.

2. **Improvement of Topology Optimization tool (chapter 4):** the improvement of the TO tool (chapter 4) remains an important area for future work. The computational cost of large scale TO can be considerable, particularly when applied to detailed magnet geometries. Enhancing the efficiency and speed of the TO framework through strategies such as parallelization, model order reduction, or the use of more efficient solvers represents a key step toward making the methodology practical for large scale problems and enabling faster design iterations while reducing overall computational effort. In addition, the introduction of manufacturing constraints that guide material removal could be further investigated, as these constraints provide an effective means of reducing the complexity of topology optimized structures. Finally, exploring alternative filtering approaches, beyond the current element-connectivity method used for extrusion constraints, would extend the applicability of the framework to non structured meshes and further improve

computational performance (the extrusion matrix represents a bottleneck, since its construction depends on the number of elements within the design domain and adjacency relationship).

3. **Application of Topology Optimization tool to Reactor-Scale Magnets (chapter 4):** the methodologies should be extended and validated on reactor-scale stellarator magnets, where the structural, thermal, and electromagnetic challenges are magnified. This step will be essential to assess the scalability and industrial applicability of the developed TO tool (sec. 1.6).

4. **Exploration of Alternative Reinforcement Layouts (chapter 3):** the parametric modeling demonstrated the trade-offs between inner support thickness and external ribs. Future work should expand the design space to include alternative reinforcement strategies that could balance structural performance with manufacturability. In alternative, the structural reinforcements can be inspired by the use of the developed TO tool (sec. 1.6) with additional manufacturing constraints, and then further refined with parametric models. Moreover, it may be beneficial to employ gradient-based optimization algorithms to increase the number of structural features explored within the parametric framework, thereby overcoming some of the inherent limitations of population-based methods such as PSO

5. **Thermal Topology Optimization for Cooling Systems:** future research should extend TO beyond purely structural considerations, incorporating thermal optimization to design cooling channels for the magnets. Such an approach would enable the simultaneous optimization of reinforcement structures and integrated cooling systems, ensuring both mechanical integrity and effective thermal management.

Bibliography

[1] Alejandro Diaz and Ole Sigmund. Checkerboard patterns in layout optimization. *Structural Optimization*, 10:40–45, 08 1995.

[2] James K. Guest. Imposing maximum length scale in topology optimization. *Structural and Multidisciplinary Optimization*, 37(5):463–473, 2009.

[3] Eduardo Fernandez, Maxime Collet, Pablo Alarcón, Simon Bauduin, and Pierre Duysinx. An aggregation strategy of maximum size constraints in density-based topology optimization. *Structural and Multidisciplinary Optimization*, 60, 11 2019.

[4] David Moiraf. *The use of Plasma Mirror for Relativistic Electron Generation Relevant to Fast Ignition in Inertial Confinement Fusion.* PhD thesis, 07 2020.

[5] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.

[6] Erik Holmberg, Bo Torstenfelt, and Anders Klarbring. Stress constrained topology optimization. *Struct. Multidiscip. Optim.*, 48(1):33–47, July 2013.

[7] Weiwei Wang, Hongling Ye, and Yunkang Sui. Lightweight topology optimization with buckling and frequency constraints using the independent continuous mapping method. *Acta Mechanica Solida Sinica*, 32(3):310–325, 2019.

[8] David Guirguis, Karim Hamza, Mohamed Aly, Hesham Hegazi, and Kazuhiro Saitou. Multi-objective topology optimization of multi-component continuum structures via a kriging-interpolated level set approach. *Structural and Multidisciplinary Optimization*, 51(3):733–748, 2015.

[9] Hayoung Chung, Oded Amir, and H. Alicia Kim. Level-set topology optimization considering nonlinear thermoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 361:112735, April 2020.

[10] Federico Ferrari, Ole Sigmund, and James K. Guest. Topology optimization with linearized buckling criteria in 250 lines of matlab. 2021.

[11] Gil Ho Yoon. Topology optimization for stationary fluid–structure interaction problems using a new monolithic formulation. *International Journal for Numerical Methods in Engineering*, 82(5):591–616, 2010.

[12] Ole Sigmund and Kurt Maute. Topology optimization approaches a comparative review. *Structural and Multidisciplinary Optimization*, 48, 12 2013.

[13] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9):635–654, 1999.

[14] *ANSYS Theory Reference*, 2021. Release 2021 R2.

[15] Krister Svanberg and Henrik Svärd. Density filters for topology optimization based on the pythagorean means. *Structural and Multidisciplinary Optimization*, 48(5):859–875, 2013.

[16] John A. Morelli. Evaluation of topology optimization filtering with numeric examples. Class notes for AOE 5904, 2019. Prepared for Dr. Robert Canfield, Virginia Tech, Fall 2019.

[17] Tyler Bruns and Daniel Tortorelli. Topology optimization of non-linear structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190:3443–3459, 03 2001.

[18] Pedro Fernandes, Àlex Ferrer, Paulo Teixeira, M. Parente, Ricardo Pinto, and Nuno Correia. Stress-constrained topology optimization for commercial software: A python implementation for abaqus. *Applied Sciences*, 13:12916, 12 2023.

[19] Alok Sutradhar, Jaejong Park, Payam Haghighi, Duane Detwiler, and Jami Shah. Incorporating manufacturing constraints in topology optimization methods: A survey. page V001T02A073, 08 2017.

[20] Sandro Vatanabe, Tiago Lippi, Cicero R. de Lima, Glaucio Paulino, and Emílio Silva. Topology optimization with manufacturing constraints: A unified projection-based approach. *Advances in Engineering Software*, 100:97–112, 10 2016.

[21] Bo Wang, Yan Zhou, Kuo Tian, and G.M. Wang. Novel implementation of extrusion constraint in topology optimization by helmholtz-type anisotropic filter. *Structural and Multidisciplinary Optimization*, 62, 10 2020.

[22] Jiaming Ma, Yunzhen He, Zi-Long Zhao, and Yi Min Xie. Topology optimization of ribbed slabs and shells. *Engineering Structures*, 277:115454, 2023.

[23] Simone Del Nero, Pierluigi Fanelli, Victor Prost, and Stefano Meloni. Element-based connectivity approach for topology optimization of rib layout design. Manuscript submitted for peer review, 2025.

[24] Sai C. Tadepalli, Ahmet Erdemir, and Peter R. Cavanagh. Comparison of hexahedral and tetrahedral elements in finite element analysis of the foot and footwear. *Journal of Biomechanics*, 44(12):2337–2343, 2011.

[25] Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

[26] Alexander Kaszynski. pyansys: Pythonic interface to MAPDL, November 2021.

[27] Pymapdl topology optimization code for 3d printed structures. `https://github.com/mid2SUPAERO/SOMP_Ansys`. Accessed March 2025.

[28] P. Fernandes, R. F. R. Pinto, M. P. L. Parente, and N. Correia. Python code for stress constrained topology optimization in abaqus. `https://github.com/pnfernandes/Python-Code-for-Stress-Constrained-Topology-Optimization-in-ABAQUS`, 2023. Accessed March 2025.

[29] Arjen Deetman. Gcmma-mma-python: Python implementation of the method of moving asymptotes. `https://github.com/arjendeetman/GCMMA-MMA-Python`, 2024. Accessed March 2025.

[30] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski. Particle swarm optimization. *AIAA Journal*, 41(8):1583–1589, 2003.

[31] R.E. Perez and K. Behdinan. Particle swarm approach for structural design optimization. *Computers Structures*, 85(19):1579–1588, 2007.

[32] Waqas Haider Bangyal, Abdul Hameed, Wael Alosaimi, and Hashem Alyami. A new initialization approach in particle swarm optimization for global optimization problems. *Computational Intelligence and Neuroscience*, 2021(1):6628889, 2021.

[33] Sabine Helwig, Juergen Branke, and Sanaz Mostaghim. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 17(2):259–271, 2013.

[34] Nikhil Padhye, Kalyanmoy Deb, and Pulkit Mittal. Boundary handling approaches in particle swarm optimization. In Jagdish Chand Bansal, Pramod Kumar Singh, Kusum Deep, Millie Pant, and Atulya K. Nagar, editors, *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, pages 287–298, India, 2013. Springer India.

[35] Joaquim Martins and Andrew Ning. *Engineering Design Optimization*. 10 2021.

[36] University of Alberta. Ansys tutorials.

[37] Marco Ariola and Alfredo Pironti. *Magnetic Control of Tokamak Plasmas*. Advances in Industrial Control. Springer, Cham, 2nd edition, 2016. Softcover reprint 2018, ISBN 9783319806792.

[38] Renaissance fusion official website, https://renfusion.eu. Accessed: June 2025.

[39] Jihong Zhu, Weihong Zhang, and Liang Xia. Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering*, 04 2015.

[40] Onyekachi Okorie, Asma Perveen, Didier Talamona, and Konstantinos Kostas. Topology optimization of an aerospace bracket: Numerical and experimental investigation. *Applied Sciences*, 13(24), 2023.

[41] K. D. Tsavdaridis. Applications of topology optimization in structural engineering: High-rise buildings and steel components. *Jordan Journal of Civil Engineering*, 9(3):335–357, July 2015.

[42] Hitoshi Tamura, Takuya Goto, Junichi Miyazawa, Teruya Tanaka, and Nagato Yanagi. Seismic analysis of magnet systems in helical fusion reactors designed with topology optimization. *IEEE Transactions on Applied Superconductivity*, 32(6):1–4, 2022.

[43] Caoxiang Zhu, Kenneth Hammond, Thomas Brown, David Gates, Michael Zarnstorff, Keith Corrigan, Marc Sibilia, and Eliot Feibush. Topology optimization of permanent magnets for stellarators. *Nuclear Fusion*, 60(10):106002, aug 2020.

[44] Oliver Marshall, Lee Margetts, L. E., and M. Roy. Thermal topology optimisation of a plasma facing component for use in next-generation fusion reactors. 04 2021.

[45] Bailey Curzadd, Alexander von Müller, Rudolf Neu, and Udo von Toussaint. Topology optimization of tungsten/copper structures for plasma-facing component applications. *Nuclear Fusion*, 59(8):086003, jun 2019.

[46] M. Baelmans, M. Blommaert, W. Dekeyser, and T. Van Oevelen. Achievements and challenges in automated parameter, shape and topology optimization for divertor design. *Nuclear Fusion*, 57(3):036022, jan 2017.

[47] M. Iguchi, T. Sakurai, M. Nakhira, N. Koizumi, and H. Nakajima. Cryogenic structural materials of the iter toroidal field coil structure. In *Fourth International*

Workshop on Structural Materials for Innovative Nuclear Systems (SMINS-4), Paris, France, 2016. Nuclear Energy Agency.

[48] Felix Schauer, Konstantin Egorov, and Victor Bykov. Helias 5-b magnet system structure and maintenance concept. *Fusion Engineering and Design*, 88(9):1619–1622, 2013. Proceedings of the 27th Symposium On Fusion Technology (SOFT-27); Liège, Belgium, September 24-28, 2012.

[49] J. Lion, J.-C. Anglès, L. Bonauer, A. Bañón Navarro, S.A. Cadena Ceron, R. Davies, M. Drevlak, N. Foppiani, J. Geiger, A. Goodman, W. Guo, E. Guiraud, F. Hernández, S. Henneberg, R. Herrero, C. Hintze, H. Höchter, J. Jelonnek, F. Jenko, R. Jorge, M. Kaiser, M. Kubie, E. Lascas Neto, H. Laqua, M. Leoni, J.F. Lobsien, V. Maurin, A. Merlo, D. Middleton-Gear, M. Pascu, G.G. Plunk, N. Riva, M. Savtchouk, F. Sciortino, J. Schilling, J. Shimwell, A. Di Siena, R. Slade, T. Stange, T.N. Todd, L. Wegener, F. Wilms, P. Xanthopoulos, and M. Zheng. Stellaris: A high-field quasi-isodynamic stellarator for a prototypical fusion power plant. *Fusion Engineering and Design*, 214:114868, 2025.

[50] F. Schauer, H. Wobig, K. Egorov, V. Bykov, and M. Köppen. Extrapolation of the w7-x magnet system to reactor size. *Contributions to Plasma Physics*, 50(8):750–755, 2010.

[51] Max-Planck-Institute for Plasma Physics. Wendelstein 7-x. https://www.ipp.mpg.de/w7x. Accessed: September 2025.

[52] Yifei Zhang. REBCO tape production at SuperPower: state of the art and prospects. 1st High Temperature Superconductor Accelerator Technology (HiTAT) Workshop, CERN, March 2023. Presented on March 9, 2023.

[53] Hamed Milanchian, Shahriar Bakrani Balani, and Tiina Salmi. Mechanical simulation of 2g hts tapes and stacks during localized temperature increase. *IEEE Transactions on Applied Superconductivity*, 35(5):1–6, 2025.

[54] R.A. Schapery. Thermal expansion coefficients of composite materials based on energy principles. *Journal of Composite Materials*, 2(3):380–404, 1968.

[55] Wenhai Zhou, Bingxu Su, Rongli Jia, and Jiafeng Cao. Electro-magnetic-mechanical characteristics of 3d hts ins and ni racetrack coils in complex physical fields. *Cryogenics*, 145:104003, 2025.

[56] Yaning Wang and Ze Jing. Multiscale modelling and numerical homogenization of the coupled multiphysical behaviors of high-field high temperature superconducting magnets. *Composite Structures*, 313:116863, 2023.

[57] Kai Liu and Andrés Tovar. An efficient 3d topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 50(6):1175–1196, 2014.

[58] *ANSYS Element Reference*, 2021. Release 2021 R2.

[59] H. Grandin. *Fundamentals of The Finite Element Method.* Macmillan Publishing Company, New York, 1986.

[60] C.S. Krishnamoorthy. *Finite Element Analysis: Theory and Programming.* Tata McGraw-Hill Publishing Company, New Delhi, 1987.

[61] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method, Vol. I & II.* McGraw Hill Book Company, London, 1989.

[62] The engineering toolbox, https://www.engineeringtoolbox.com. Accessed: June 2025.

.1 HTS Equivalent Mechanical Properties: Homogenization Technique

High-Temperature Superconducting (HTS) tapes are composite materials consisting of multiple layers with distinct mechanical properties. Directly modeling each layer within structural simulations is computationally expensive and often impractical. To address this, a homogenization technique is employed to derive equivalent mechanical properties that approximate the behavior of the layered structure as a single, homogeneous material. This approach simplifies the analysis while preserving the overall stiffness and strength characteristics of the HTS stack. The following describes the method used to compute the effective elastic properties based on the volume fraction and properties of the individual constituents.

Assuming the HTS conductor is composed of n distinct layers, the effective mechanical properties in a cylindrical coordinate system placed on HTS center, can be determined using homogenization techniques. The homogenized parameters include:

- Radial E_r, tangential E_θ, and axial E_z Young's moduli;
- In-plane $G_{r\theta}$ and out-of-plane $G_{\theta z}$ shear moduli;
- In-plane $\nu_{r\theta}$ and out-of-plane $\nu_{\theta z}$.

The homogenized properties are expressed as functions of the thickness fraction of each material layer, $x_i = \frac{t_i}{t_{tot}}$, where t_i is the thickness of layer i, and t_{tot} is the total thickness of the HTS stack.

The homogenized properties are calculate using the following equations:

$$E_r = \left(\sum_{i=1}^{n} \frac{x_i}{E_i} \right)^{-1} \tag{1}$$

$$E_\theta = E_z = \sum_{i=1}^{n} E_i \cdot x_i \tag{2}$$

$$G_{r\theta} = \left(\sum_{i=1}^{n} \frac{x_i}{G_{r\theta}(i)} \right)^{-1} \tag{3}$$

$$G_{\theta z} = \sum_{i=1}^{n} G_{\theta z}(i) \cdot x_i \tag{4}$$

$$\nu_{r\theta} = \sum_{i=1}^{n} \nu_{r\theta}(i) \cdot x_i \tag{5}$$

$$\nu_{\theta z} = \frac{\sum_{i=1}^{n} \nu_{\theta z}(i) \cdot E_i \cdot x_i}{\sum_{i=1}^{n} E_i \cdot x_i} \tag{6}$$

.2 High level calculation for magnet reinforcement segmentation

In this appendix, a simplified analytical method is presented to estimate the required number of bolts in the case where the magnet reinforcement structure, described in chapter 4, is split into two equal segments. The purpose of this calculation is to provide a high-level sizing criterion, allowing an early-stage assessment of the feasibility of segmented reinforcements.

The bolts located along the joint must carry the separating loads generated by the magnetic forces. These loads can be categorized as:

- **Normal force (separating load)**, acting perpendicular to the joint plane and responsible for pulling the two segments apart;
- **Tangential force (shear load)**, acting in-plane along the joint and resisted through shear capacity of the bolts.

Two approaches may be considered to evaluate the joint loads:

1. Direct integration of the pressure load over the reinforcement surface;
2. Evaluation from a numerical model, which is more practical and quicker to implement.

In this work, the separating loads were derived numerically by interpolating the EM loads shown in sec. 4.2, and decoupling the interaction between the two segments with two separate fixed constraints (see Fig. 1b). The resulting axial separating force is:

$$F_{\text{separating}} \approx 21.3\,\text{MN}$$

acting on the upper joint, and an equal and opposite force of

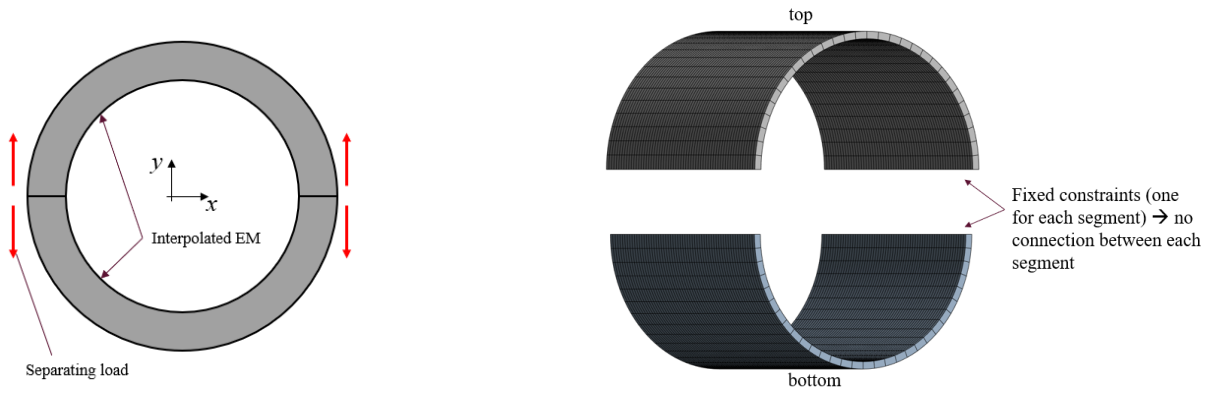
$$F_{\text{separating}} \approx -21.3\,\text{MN}$$

acting on the lower joint.

For the bolt-sizing calculation, the following assumptions are made:

- Magnet axial length: $L = 0.52$ m;
- Minimum distance between bolts: $\Delta s \approx 2 \cdot d_{\text{bolt}}$, where d_{bolt} is the bolt diameter;

- The reinforcement structure is modeled as a solid cylinder (see Fig.1) split into two equal segments.



(a) (b) .

Figure 1: Representation of the segmented reinforcement: (a) schematic cross-section highlighting the separating loads acting on the joint; (b) finite element model of the separated reinforcement showing the location of fixed constraints.

Based on these assumptions, the reinforcements can be thought of as being split into two "omega-shaped" segments (see schematic view in Fig. 2). At the interface, the joints are flat plates where bolts are arranged to connect the two segments. These flat joints (or radial plates) are assumed to be infinitely stiff; however, a dedicated FE model with the radial plates and bolts could be implemented to verify the accuracy of this high-level analytical estimation. The detailed results, including the number of bolts required for different bolt diameters, are summarized in Table 1. Here the type of bolts were considered according to [62].

Table 1: Results of bolt number estimation for magnet reinforcement segmentation.

Bolt (class 12.9)	F_{max} [kN]	Δs [mm]	$n_{min,bolts}$	$n_{max,bolts,row}$	$n_{rows,side}$	$n_{bolts,used}$	Plate radial length [mm]	SF
M12	103	24	208	23	5	230	120	1.11
M18	234	36	92	14	4	120	144	1.32
M24	431	48	50	11	3	72	144	1.46
M30	684	60	32	9	2	36	120	1.16

In Table 1, the meaning of each column is detailed below:

- F_{max}: maximum load that a single bolt can sustain;
- Δs: minimum spacing between adjacent bolts, generally assumed as twice the bolt diameter;
- $n_{min,bolts}$: minimum number of bolts required to carry the total separating load, computed as:

$$n_{min,bolts} = \frac{F_{separating}}{F_{max}}$$

- $n_{max,bolts,row}$: maximum number of bolts that can be arranged along a single row, given the available plate length and spacing constraints (see Fig. 2), computed as the higher integer as following:

$$n_{max,bolts,row} = \text{int}\left[\frac{L}{n_{min,bolts}}\right]$$

- $n_{rows,side}$: number of bolt rows that can be accommodated on each side of the joint (see Fig. 2), computed as the higher integer as following:

$$n_{rows,side} = \text{int}\left[\frac{n_{min,bolts}}{n_{max,bolts,row} \cdot 2}\right] \cdot 2$$

- $n_{bolts,used}$: actual number of bolts selected for the configuration, computed as the higher integer as following:

$$n_{bolts,used} = \text{int}\left[n_{rows,side} \cdot n_{max,bolts,row}\right] \cdot 2$$

- Plate radial length: radial dimension of the joint plate considered in the analysis (see Fig. 2), evaluated as:

$$\text{Plate radial length} = \frac{\Delta s \cdot n_{rows,side}}{2}$$

- SF: safety factor associated with the chosen configuration, evaluated as:

$$\text{SF} = \frac{F_{max}}{F_{separating}/n_{bolts,used}}$$

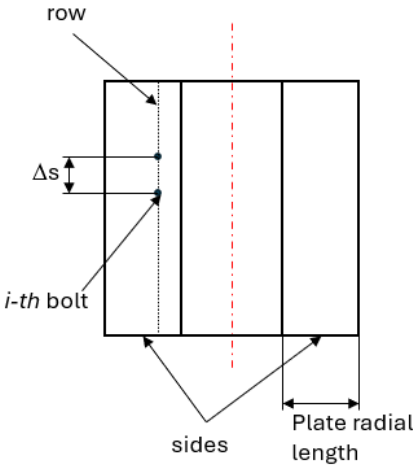


Figure 2: Simplified top view of the segmented reinforcement joint, illustrating the bolt arrangement, bolt rows, plate radial length, sides, and bolt spacing Δs.

The results reported in Table 1 allow us to compare different bolt sizes in terms of spacing requirements, SF and so on. A few observations can be highlighted:

- **M12 bolts:** the smallest bolt size considered requires the highest number of bolts (230 in total). Although the SF = 1.11 is acceptable, the large number of bolts makes this solution impractical from both a manufacturing and assembly point of view;
- **M18 bolts:** the number of bolts is reduced to 144, while the SF increases to 1.32. This represents a more efficient compromise compared to M12 bolts, though the overall number of bolts is still relatively high;
- **M24 bolts:** this configuration also requires 144 bolts but with a further improvement in SF (SF = 1.46). The reduced number of rows per side (3 instead of 4) and shorter plate radial length (72 mm) make this solution attractive from both structural and practical perspectives;
- **M30 bolts:** The minimum number of bolts (120) is achieved with the largest bolt size. However, the SF (SF = 1.16) is lower than for M18 and M24 bolts.

The addition of orthogonal joints to the cylindrical surface of the magnet reinforcement adds further dimensions to the structure. In fact, if there is a central ring, these joints cannot extend towards the center of the machine as they would prevent the central ring from coupling with the magnet reinforcement itself.

.3 Tapered Fit Interference

The purpose of this appendix is to develop a simplified analytical model for a tapered interference fit, and to compare its predictions with a numerical solution obtained using ANSYS Workbench. The aim is not to provide a detailed design tool, but rather to derive a model capable of delivering first-order estimates that can be used during preliminary design stages.

Objectives

The analytical formulation is intended to:

- provide an order-of-magnitude estimate of the contact pressure generated, as well as the radial and tangential stresses, at the interface as a function of the main governing parameters, including the imposed interference (f), cylinder radii ($a,\ b_{min},\ b_{max},\ c$), taper angle ($\alpha$), friction coefficient (μ), and material properties ($E_i,\ \nu_i,\ E_o,\ \nu_o$);
- estimate the peak axial force (F_z) required to slide the inner cylinder within the outer one.

Analytical formulation

When two cylindrical components are assembled by shrinking or press fitting one part upon another, a contact pressure (p_f) is developed between the two parts. The contact pressure at the nominal radius b is as following (see Fig. 3):

$$p_f = \frac{f}{b \cdot k} \tag{7}$$

where k is a parameters that consider cylinder geometries and their mechanical properties, and it is defined as:

$$k = \frac{1}{E_o}\left(\frac{c^2 + b^2}{c^2 - b^2} + \nu_o\right) + \frac{1}{E_i}\left(\frac{b^2 + a^2}{b^2 - a^2} - \nu_i\right) \tag{8}$$

In Eq. 8, the subscript i denotes the inner cylinder, while o the outer one.

Furthermore, what can be observed in Eq. 7, for straight cylindrical interference surface, the contact pressure depends on:

1. interference between the two parts;

2. geometrical parameters of the two parts (inner, outer and nominal radii);

3. mechanical properties of the two parts.

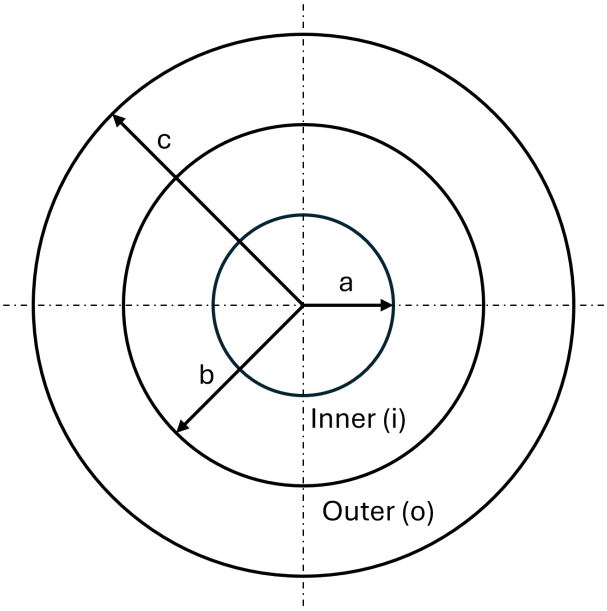


Figure 3: Cross-section of press-fitted straight cylinders: a : inner radius; b : radius at interference; c : outer radius.

In case of conical interference surface, the nominal radius is no more constant along the axial direction (z-axis) and needs to be evaluated at each axial coordinate (see Fig. 4). Here, the advantage lies on avoiding to heating up or cooling down one of the two components. Considering a cylindrical coordinate system where z points along the axial direction (axisymmetric axis) and r is the radial coordinate, the nominal radius b can be expressed as:

$$b = b_{min} + (L - z)\tan(\alpha) \tag{9}$$

where b_{min} is the minimum tapered radius.

To compute the net force to be applied on the inner cylinder such that it will slide inside the external one, two terms must be evaluated:

- the contribution due to axial force of the contact pressure:

$$dF_{normal} = p_f \; dA \rightarrow dF_{z,1} = p_f \; \sin(\alpha) \; dA \tag{10}$$

- the contribution due to the friction contact between the two components:

$$dF_{friction} = \mu \; p_f \; dA \rightarrow dF_{z,2} = \mu \; p_f \; \cos(\alpha) \; dA \tag{11}$$

Therefore, the infinitesimal axial force to be applied to slide the inner cylinder is:

$$dF_{z,tot} = p_f \ (\sin(\alpha) + \mu \cos(\alpha)) \ dA \tag{12}$$

The infinitesimal area dA in Eq. 10, 11 and 12 is a a function of the axial coordinate, and is defined as:

$$dA = b(z) \ d\theta \ ds = b(z) \ d\theta \ \frac{dz}{\cos(\alpha)} \tag{13}$$

Inserting Eq. 13 in 12, we finally get the total axial force:

$$F_{z,tot} = \int_0^{2\pi} d\theta \int_{b_{max}}^{b_{min}} p_f(z) \ \ (\tan(\alpha) + \mu)) \ \ b(z) \ dz \tag{14}$$

The stress generated at the interface between the two cylinder are radial ($\sigma_{r,i}, \ \sigma_{r,o}$) and tangential ($\sigma_{t,i}, \ \sigma_{t,o}$):

$$\sigma_{r,i} = -\frac{p_f \ b^2}{b^2 - a^2} \left(1 - \frac{a^2}{r^2}\right) \tag{15}$$

$$\sigma_{r,o} = \frac{p_f \ b^2}{c^2 - b^2} \left(1 - \frac{c^2}{r^2}\right) \tag{16}$$

$$\sigma_{t,i} = -\frac{p_f \ b^2}{b^2 - a^2} \left(1 + \frac{a^2}{r^2}\right) \tag{17}$$

$$\sigma_{t,o} = \frac{p_f \ b^2}{c^2 - b^2} \left(1 + \frac{c^2}{r^2}\right) \tag{18}$$

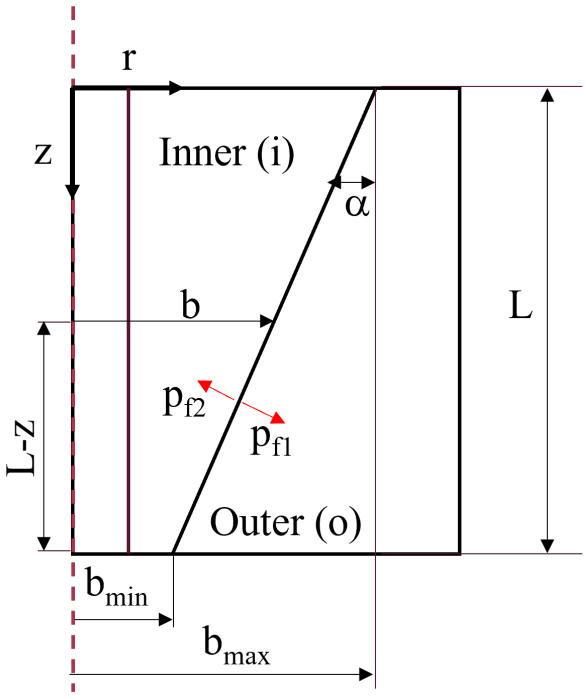


Figure 4: Cross-section of tapered fit: r: radial coordinate; z : axial coordinate; L : axial length; b : tapered radius; α: tapered angle; b_{min}: minimum tapered radius; b_{max}: maximum tapered radius; p_{f1}: contact pressure on the outer cylinder; p_{f2}: contact pressure on the inner cylinder.

Validation with FE simulations

To assess the accuracy of the analytical model, Eq. 7, 15, 16, 17 and 18 were compared with results from a simple FE analysis carried out in ANSYS Workbench.

The FE model is constructed using two quarter-cylinder sections discretized with SOLID185 elements (see Fig. 5). Nonlinear contact elements are introduced at the interface to capture the interference fit behavior. Boundary conditions are applied such that one side of the model is constrained in the vertical direction, while the opposite side is restrained in the horizontal direction, ensuring overall stability while allowing deformation consistent with the physical assembly.

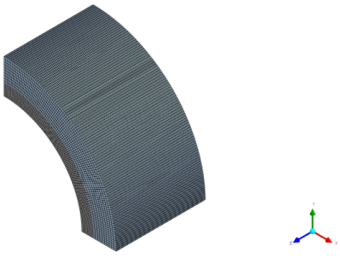


Figure 5: FE model of two cylinder connected with tapered fit interference.

f	L	α	a	c	$b_{\min}$	$b_{\max}$	$E_i = E_o$	$\nu_i = \nu_o$	μ
0.01 mm	0.5321 m	0.1°	0.465 m	0.67 m	0.5300 m	0.5309 m	200 GPa	0.3	0.15

Table 2: Parameters used in FE model: f : interference; L : axial length; α: tapered angle; a: minimum inner cylinder radius; c: maximum outer cylinder radius; b_{min}: minimum tapered radius; b_{max}: maximum tapered radius; $E_i = E_o$: Young's modulus of inner and outer cylinders; $\nu_i = \nu_o$: Poissoin's ratio of inner and outer cylinders.

In Fig. 6, 7 and 8 are shown the numerical results of radial stress, tangential stress and pressure contact (pressure contact evaluated at the middle plane), respectively. In more detail, results are grabbed at the interface and compared with the analytical one (see Table 3). The analytical model provides results that are in very good agreement with the FE solution:

- the contact pressure P_c differs by less than 1.3% (0.315 MPa vs. 0.319 MPa);
- the radial stresses at the inner and outer surfaces ($\sigma_{r,i}$ and $\sigma_{r,e}$) show a deviation of about 3.8%;
- for the tangential stress at the inner surface ($\sigma_{t,i}$), the deviation is larger, around 18.5%, which might be attributed the simplified analytical formulation;
- the tangential stress at the outer surface ($\sigma_{t,e}$) shows a discrepancy of about 5.7%, still within an acceptable range for preliminary design calculations.

Overall, the comparison demonstrates that the analytical approach provides a reliable order-of-magnitude estimate of the contact pressure and stress distribution. While local stresses—particularly $\sigma_{t,i}$—are more sensitive to modeling assumptions, the analytical method remains a useful tool for quick parametric evaluations before resorting to detailed FE simulations.

Finally, for the analyzed case the peak force required to slide the inner cylinder within the outer one is

$$F_{z,tot} = 78.23 \text{ kN}$$

The pressure to be applied on the inner cylinder in order to slide it within the outer is:

$$P_z = \frac{F_{z,tot}}{A_i} = \frac{a\ F_{z,tot}}{\pi\ (b_{max}^2 - a^2)} = 1.64 \text{ MPa} \tag{19}$$

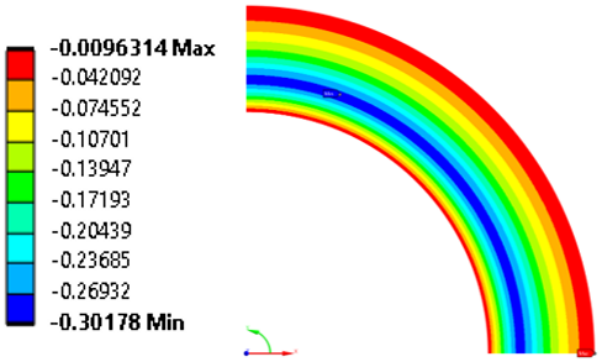


Figure 6: Radial stress σ_r [MPa].

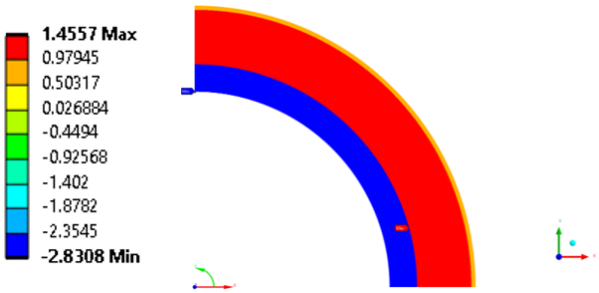


Figure 7: Tangential stress σ_t [MPa].

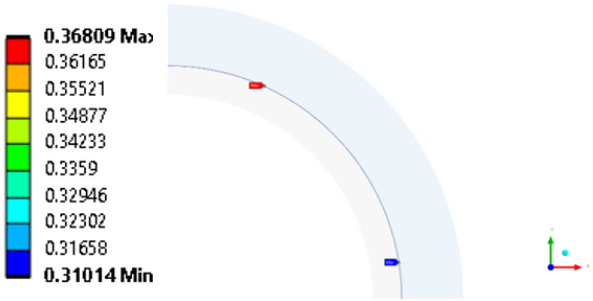


Figure 8: Pressure contact p_f at the interface [MPa].

Table 3: Comparison between analytical and numerical results for the tapered fit interference case (at the interference radius b): p_f: contact pressure; $\sigma_{r,i}$: radial stress on inner cylinder; $\sigma_{r,o}$: radial stress on outer cylinder; $\sigma_{t,i}$: tangential stress on inner cylinder; $\sigma_{t,o}$: tangential stress on outer cylinder.

	Analytical	Numerical
p_f [MPa]	0.315	0.319
$\sigma_{r,i}$ [MPa]	-0.314	-0.302
$\sigma_{r,e}$ [MPa]	-0.314	-0.302
$\sigma_{t,i}$ [MPa]	-2.389	-2.831
$\sigma_{t,e}$ [MPa]	1.378	1.456

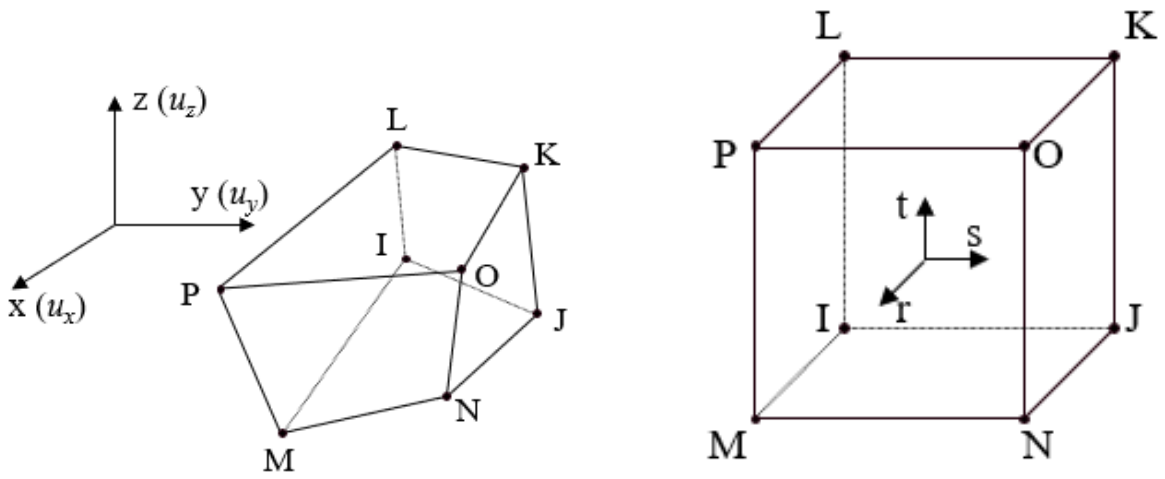
.4 8-node brick element

One of the most difficult challenges in stress-constrained TO is the sensitivity analysis. As discussed, the stress gradient requires the solution of an adjoint problem, which leads to the implementation of very low-level FE formulations. In particular, the matrix that bridges the nodal displacements $\mathbf{u}$ with the associated nodal strain $\boldsymbol{\varepsilon}$, commonly referred to as the strain-displacement matrix $\mathbf{B}$, becomes central.

In this work, the $\mathbf{B}$ matrix of SOLID185 elements was analyzed and compared with an in-house PyMAPDL implementation. The aim of this appendix is to provide both a theoretical and practical approach to the implementation of this matrix, as well as the elemental stiffness matrix $\mathbf{k}_e$, and the main differences obtained using in-house code for the construction of such low level FE matrices.

.4.1 Element Formulation

The 8-node brick element is widely used in structural mechanics due to its balance between computational cost and accuracy. It is composed of 8 nodes forming a hexahedron, which may not be a perfect cube in general (see Fig.9). Each node has 3 translational degrees of freedom (dof) in x, y and z directions, making the total dof of this element to be 24.



(a) (b)

Figure 9: Isoparametric 8-node brick element: (a) element in the physical space, (b) element in the natural space.

The displacement within the element are described by the shape functions and nodal displacements:

$$u_x(x,y,z) = \sum_{i=1}^{8} N_i(r,s,t) \cdot u_{x,i} \tag{20}$$

$$u_y(x,y,z) = \sum_{i=1}^{8} N_i(r,s,t) \cdot u_{y,i} \tag{21}$$

$$u_z(x,y,z) = \sum_{i=1}^{8} N_i(r,s,t) \cdot u_{z,i} \tag{22}$$

In analogous manner, when using an isoparametric element, the physical coordinates (x, y, z) of any point within the element are interpolated using the same shape functions in Eq. 20, 21 and 22:

$$x = \sum_{i=1}^{8} N_i(r,s,t) \cdot x_i \tag{23}$$

$$y = \sum_{i=1}^{8} N_i(r,s,t) \cdot y_i \tag{24}$$

$$z = \sum_{i=1}^{8} N_i(r,s,t) \cdot z_i \tag{25}$$

In isoparametric formulation, as we shown above, the geometry and the displacement field of the element are described by the same set of shape functions $\mathbf{N}$. The element, in the natural coordinate system $r - s - t$, has side equal to 2 (the lengths facilitate the use of standard Gauss integration, since the the reference system is at the element centroid). Each shape function is associated with a node located at one of the corners of the element.

For a typical node ordering (ANSYS SOLID185 element), the local coordinates are as follows:

All shape functions for the 8-node hexahedral element can be compactly represented using the local (natural) coordinates of each node, denoted as r_i, s_i, t_i for $i = 1, \ldots, 8$, corresponding to the nodal positions in the transformed (parent) element domain (see Table 4).

$$N_i(r,s,t) = \frac{1}{8}(1 + r \cdot r_i)(1 + s \cdot s_i)(1 + t \cdot t_i) \quad \text{for } i = 1, \ldots, 8 \tag{26}$$

The shape functions, as highlighted in Eq. 26, for the 8-node brick element are trilinear.

Node	r_i	s_i	t_i
I (1)	-1	-1	-1
J (2)	-1	-1	1
K (3)	-1	1	1
L (4)	-1	1	-1
M (5)	1	-1	-1
N (6)	1	-1	1
O (7)	1	1	1
P (8)	1	1	-1

Table 4: ANSYS node ordering for the SOLID185 element in the (r, s, t) natural coordinate system.

Expanding them, the shape function vector is composed of the following:

$$N_1(r,s,t) = \frac{1}{8}(1-r)(1-s)(1-t) \tag{27}$$

$$N_2(r,s,t) = \frac{1}{8}(1+r)(1-s)(1-t) \tag{28}$$

$$N_3(r,s,t) = \frac{1}{8}(1+r)(1+s)(1-t) \tag{29}$$

$$N_4(r,s,t) = \frac{1}{8}(1-r)(1+s)(1-t) \tag{30}$$

$$N_5(r,s,t) = \frac{1}{8}(1-r)(1-s)(1+t) \tag{31}$$

$$N_6(r,s,t) = \frac{1}{8}(1+r)(1-s)(1+t) \tag{32}$$

$$N_7(r,s,t) = \frac{1}{8}(1+r)(1+s)(1+t) \tag{33}$$

$$N_8(r,s,t) = \frac{1}{8}(1-r)(1+s)(1+t) \tag{34}$$

where they can be expressed in a compact form:

$$\mathbf{N} = [N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8] \tag{35}$$

where u_x, u_y and u_z are the displacement along x, y and z of any point within the element, while $u_{x,i}$, $u_{y,i}$ and $u_{z,i}$ correspond to the nodal displacements along the three directions.

Since the internal strains ($\boldsymbol{\varepsilon}$) at a point are derivatives of the displacement ($\mathbf{u}$) w.r.t. the x, y and z axes, and displacement are expressed in $r-s-t$ coordinates, a transformation between the physical space and the natural coordinate systems is required. The transformation from the cartesian to the natural coordinate system occurs through the Jacobian matrix $\mathbf{J}$. The role of $\mathbf{J}$, as mentioned, consist of mapping derivatives from the natural coordinate space (r, s, t) to the physical coordinate space (x, y, z). This is

transformation is defined as:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{bmatrix} \tag{36}$$

where $\mathbf{J}$ is defined as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \tag{37}$$

The Jacobian matrix contains the partial derivative of nodal coordinates in the natural space. Considering that the element has 8 nodes, and the nodal coordinates are expressed as in Eq. 23, 24 and 25, $\mathbf{J}$ becomes:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \cdots & \frac{\partial N_8}{\partial r} \\ \frac{\partial N_1}{\partial s} & \frac{\partial N_2}{\partial s} & \cdots & \frac{\partial N_8}{\partial s} \\ \frac{\partial N_1}{\partial t} & \frac{\partial N_2}{\partial t} & \cdots & \frac{\partial N_8}{\partial t} \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_8 & y_8 & z_8 \end{bmatrix} = \nabla \mathbf{N} \cdot \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_8 & y_8 & z_8 \end{bmatrix} \tag{38}$$

For an 8-node brick element, the derivatives of the shape functions with respect to the natural coordinates $r, s, t \in [-1, 1]$ are given by:

$$\begin{aligned}
\frac{\partial N_1}{\partial r} &= -\frac{1}{8}(1-s)(1-t) & \frac{\partial N_2}{\partial r} &= \frac{1}{8}(1-s)(1-t) & \frac{\partial N_3}{\partial r} &= \frac{1}{8}(1+s)(1-t) \\
\frac{\partial N_4}{\partial r} &= -\frac{1}{8}(1+s)(1-t) & \frac{\partial N_5}{\partial r} &= -\frac{1}{8}(1-s)(1+t) & \frac{\partial N_6}{\partial r} &= \frac{1}{8}(1-s)(1+t) \\
\frac{\partial N_7}{\partial r} &= \frac{1}{8}(1+s)(1+t) & \frac{\partial N_8}{\partial r} &= -\frac{1}{8}(1+s)(1+t)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial N_1}{\partial s} &= -\frac{1}{8}(1-r)(1-t) & \frac{\partial N_2}{\partial s} &= -\frac{1}{8}(1+r)(1-t) & \frac{\partial N_3}{\partial s} &= \frac{1}{8}(1+r)(1-t) \\
\frac{\partial N_4}{\partial s} &= \frac{1}{8}(1-r)(1-t) & \frac{\partial N_5}{\partial s} &= -\frac{1}{8}(1-r)(1+t) & \frac{\partial N_6}{\partial s} &= -\frac{1}{8}(1+r)(1+t) \\
\frac{\partial N_7}{\partial s} &= \frac{1}{8}(1+r)(1+t) & \frac{\partial N_8}{\partial s} &= \frac{1}{8}(1-r)(1+t)
\end{aligned}$$

$$\frac{\partial N_1}{\partial t} = -\frac{1}{8}(1-r)(1-s) \quad \frac{\partial N_2}{\partial t} = -\frac{1}{8}(1+r)(1-s) \quad \frac{\partial N_3}{\partial t} = -\frac{1}{8}(1+r)(1+s)$$
$$\frac{\partial N_4}{\partial t} = -\frac{1}{8}(1-r)(1+s) \quad \frac{\partial N_5}{\partial t} = \frac{1}{8}(1-r)(1-s) \quad \frac{\partial N_6}{\partial t} = \frac{1}{8}(1+r)(1-s)$$
$$\frac{\partial N_7}{\partial t} = \frac{1}{8}(1+r)(1+s) \quad \frac{\partial N_8}{\partial t} = \frac{1}{8}(1-r)(1+s)$$

The derivatives of the shape functions can also be expressed using the following compact form:

$$\frac{\partial N_i}{\partial r} = \frac{1}{8} \cdot r_i \cdot (1 + s \cdot s_i)(1 + t \cdot t_i) \tag{39}$$

$$\frac{\partial N_i}{\partial s} = \frac{1}{8} \cdot s_i \cdot (1 + r \cdot r_i)(1 + t \cdot t_i) \tag{40}$$

$$\frac{\partial N_i}{\partial t} = \frac{1}{8} \cdot t_i \cdot (1 + r \cdot r_i)(1 + s \cdot s_i) \tag{41}$$

In FEA, the strain vector at point a is linked to the displacement through the strain-displacement matrix as:

$$\boldsymbol{\varepsilon}_a = \mathbf{B}_a \, \mathbf{u} \tag{42}$$

where $\mathbf{u}$ is the nodal displacement of the considered element, computed with Eq. 20, 21 and 22.

The strain vector is composed of 6 independent components, representing the volumetric and tangential deformations:

$$\boldsymbol{\varepsilon}_a = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_z}{\partial z} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_x}{\partial y} \\ \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} \\ \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial z} \end{bmatrix} \tag{43}$$

Expanding Eq. 43 and expressing the nodal element vector in the natural coordinate

system using the Jacobian, we get:

$$\boldsymbol{\varepsilon}_a = \mathbf{A} \begin{bmatrix} \mathbf{J^{-1}} & & \\ & \mathbf{J^{-1}} & \\ & & \mathbf{J^{-1}} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} u_{x,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} u_{x,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} u_{x,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} u_{y,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} u_{y,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} u_{y,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} u_{z,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} u_{z,i} \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} u_{z,i} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{J^{-1}} & & \\ & \mathbf{J^{-1}} & \\ & & \mathbf{J^{-1}} \end{bmatrix} \mathbf{d}\ \mathbf{u} = \mathbf{B}_a\ \mathbf{u} \tag{44}$$

where in Eq. 44 $\mathbf{A}$ is a matrix that contains 0 and 1, and $\mathbf{d}$ contains the derivative of the shape functions w.r.t. $r - s - t$.

The matrix $\mathbf{B}_a$ for an 8-node brick element is 6x24, and it is finally defined as:

$$\mathbf{B}_a = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \cdots & \frac{\partial N_8}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \cdots & 0 & \frac{\partial N_8}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & \cdots & 0 & 0 & \frac{\partial N_8}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \cdots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & \cdots & 0 & \frac{\partial N_8}{\partial z} & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \cdots & \frac{\partial N_8}{\partial z} & 0 & \frac{\partial N_8}{\partial x} \end{bmatrix} \tag{45}$$

It is important to note that $\mathbf{B}_a$ is evaluated at each integration point.

Once the strain matrix $\mathbf{B}_a$ has been evaluated for all integration points, the stiffness element matrix $\mathbf{k}_e$ can be obtained by substituting $\mathbf{B}_a$ in the following:

$$\mathbf{k}_e = \int_{v_e} \mathbf{B}_a^T\ \mathbf{D}\ \mathbf{B}_a\ dv_e \tag{46}$$

Since the analytical integration 46 might be challenging, the integrals are performed using numerical integration scheme (i.e., Gauss integration scheme). For a 3D element, like the one analyzed, the volume integral is approximated as a summation over the Gauss

points in the three directions $r - s - t$:

$$\mathbf{k}_e \approx \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} \sum_{k=1}^{n_g} \mathbf{B}_a^T(r_i, s_j, t_k)\ \mathbf{D}\ \mathbf{B}_a(r_i, s_j, t_k)\ \det \mathbf{J}(r_i, s_j, t_k)\ w_i\, w_j\, w_k \tag{47}$$

where $w_i\, w_j\, w_k$ are the weighting factors, and n_g represent the number of integration points (also referred as Gauss points). Depending on the number of integration points, the weighting factors have different values as shown in Table 5. For this type of element, two integration points are sufficient.

Number of int. points	Integration point location (ξ)	Weighting factors w
1	0	2
2	$\pm\frac{1}{\sqrt{3}} \approx \pm 0.577$	1
3	$0,\ \pm\sqrt{\frac{3}{5}}$	5/9, 8/9 (center, outer)

Table 5: Gauss integration points and corresponding weighting factors in 1D. For 3D, weights are multiplied: $w_i\, w_j\, w_k$.

.4.2 B matrix – Some adjustments

Traditional isoparametric continuum elements, such as the 8-node cubic element, are often not sufficiently robust or accurate for general stress analysis, particularly in cases involving bending-dominated behavior, nearly incompressible materials, or coarse meshes [14]. One of the main challenges lies in the strain-displacement matrix, commonly referred to as the **B** matrix, which directly influences the formulation of the stiffness matrix and, consequently, the accuracy of stress and strain predictions. Indeed, it is well know that low order elements present two main type of locking: shear locking, and volumetric locking.

To address these issues, several techniques have been developed to modify the **B** matrix, improving element performance. In the ANSYS theory reference, the SOLID185 **B** is modified with the following methodologies:

- Full integration with $\bar{\mathrm{B}}$ method: it is commonly employed to alleviate volumetric locking, particularly in nearly incompressible material scenarios. It achieves this by replacing the volumetric strain at each Gauss integration point with the average volumetric strain computed over the entire element. However, the $\bar{\mathrm{B}}$ method does not address shear locking, which can significantly affect accuracy in bending-dominated problems.

- Uniform reduced integration with hourglass control: this method is another approach that helps mitigate both shear locking in bending-dominated cases and vol-

umetric locking in nearly incompressible conditions. It utilizes a single integration point for all components of the strain field, making it computationally more efficient than methods such as selective reduced integration or enhanced strain formulations, while still maintaining reasonable accuracy in many applications.

- Enhanced strain formulation: this formulation effectively addresses shear locking in bending-dominated problems and volumetric locking in nearly incompressible materials. In this approach, ANSYS introduces a set of internal, inaccessible degrees of freedom at the element level. The enhanced strain formulation is generally less computationally efficient than either the selective reduced integration or the uniform reduced integration methods.

- Simplified enhanced strain formulation: this formulation is designed to prevent shear locking in bending-dominated problems. It resembles the full enhanced strain formulation but introduces only the internal degrees of freedom required to alleviate shear locking. Since it does not include additional internal modes to address volumetric locking, this approach is not suitable for nearly incompressible materials.

The mathematical fundamentals of all the methodologies just described above, are unfortunately not documented and there is a lack of information that might help in reconstruct those very low level FE matrices. Therefore, the implemented code doesn't take into account modification of the $\mathbf{B}$ matrix.

.4.3 Python implementation of B matrix

A general method for calculating the stiffness element matrix for isoparametric elements can be defined as:

1. Define the shape functions to describe both element geometry and displacement;
2. Define the number of Gauss points in the element for stiffness integration;
3. For each Gauss point:
   (a) Get values for Gauss point in natural coordinate system;
   (b) Compute values for shape function derivatives w.r.t. natural coordinate system at current Gauss point ($\nabla \mathbf{N}$);
   (c) Calculate Jacobian $\mathbf{J}$, its inverse $\mathbf{J^{-1}}$ and its determinant $\det \mathbf{J}$;
   (d) Calculate shape functions derivatives w.r.t. global coordinate system ($x - y - z$);

(e) Calculate strain-displacement matrix $\mathbf{B}_a$;

(f) Calculate element stiffness matrix $\mathbf{k}_e$.

Initialization

```
# Import necessary library
import numpy as np

# Create Element Formulation class
class ElementFormulation:
  def __init__(self, E, ni, coords):
        """
        Initialize the Hexahedral Element with material properties and nodal
            coordinates
        """
        self.E = E # Young's modulus [Pa]
        self.ni = ni # Poisson's ratio
        self.coords = np.array(coords) # Nodal coordinates (8 nodes)
        self.D = self.constitutive_matrix() # Elasticity matrix
        # Gauss quadrature points and weights
        self.gauss_pts = np.array([-1 / np.sqrt(3), 1 / np.sqrt(3)])
        self.weights = np.array([1, 1])
```

Constitutive matrix

```
    def constitutive_matrix(self):
    """
    Compute the elastic stiffness (constitutive) matrix for an isotropic
        material
    """
    G = self.E / (2 * (1 + self.ni)) # Shear modulus [Pa]
    D1 = self.E * (1 - self.ni) / ((1 + self.ni) * (1 - 2 * self.ni)) # First
        term
    D2 = self.ni * self.E / ((1 + self.ni) * (1 - 2 * self.ni)) # Second term
    D3 = G # Shear modulus terms
    D = np.array([[D1, D2, D2, 0, 0, 0],
                  [D2, D1, D2, 0, 0, 0],
                  [D2, D2, D1, 0, 0, 0],
                  [0, 0, 0, D3, 0, 0],
                  [0, 0, 0, 0, D3, 0],
```

```
                [0, 0, 0, 0, 0, D3]])
    return D
```

Shape function derivatives (or gradient of shape functions)

```
def shape_function_derivatives(self, xi, eta, zeta):
    """
    Compute the derivatives of the shape functions
    with respect to xi, eta, and zeta (they are scalar values)
    """
    dN_dxi = np.array([
        -(1 - eta) * (1 - zeta), (1 - eta) * (1 - zeta),
         (1 + eta) * (1 - zeta), -(1 + eta) * (1 - zeta),
        -(1 - eta) * (1 + zeta), (1 - eta) * (1 + zeta),
         (1 + eta) * (1 + zeta), -(1 + eta) * (1 + zeta)
    ]) / 8

    dN_deta = np.array([
        -(1 - xi) * (1 - zeta), -(1 + xi) * (1 - zeta),
         (1 + xi) * (1 - zeta), (1 - xi) * (1 - zeta),
        -(1 - xi) * (1 + zeta), -(1 + xi) * (1 + zeta),
         (1 + xi) * (1 + zeta), (1 - xi) * (1 + zeta)
    ]) / 8

    dN_dzeta = np.array([
        -(1 - xi) * (1 - eta), -(1 + xi) * (1 - eta),
        -(1 + xi) * (1 + eta), -(1 - xi) * (1 + eta),
         (1 - xi) * (1 - eta), (1 + xi) * (1 - eta),
         (1 + xi) * (1 + eta), (1 - xi) * (1 + eta)
    ]) / 8

    grad_N = np.vstack((dN_dxi, dN_deta, dN_dzeta))
    return grad_N
```

Jacobian matrix, Jacobian determinant and inverse of Jacobian matrix

```
def jacobian_matrix(self, grad_N):
    """
    Compute the Jacobian matrix for the transformation
    from the reference element (in natural c.s.) to the real element (in global
        c.s.)
```

```
    """
    J = grad_N @ self.coords
    detJ = np.linalg.det(J)
    invJ = np.linalg.inv(J)
    return J, invJ, detJ
```

Strain-displacement matrix B

```
def compute_B_matrix(self, invJ, grad_N):
    """
    Compute the strain-displacement matrix Ba using the inverse Jacobian
    """
    dN_dxyz = invJ @ grad_N

    B = np.zeros((6, 24))
    for i in range(8):
        B[0, i*3] = dN_dxyz[0, i] # dN_i/dx (normal strain in x)
        B[1, i*3 + 1] = dN_dxyz[1, i] # dN_i/dy (normal strain in y)
        B[2, i*3 + 2] = dN_dxyz[2, i] # dN_i/dz (normal strain in z)

        B[3, i*3] = dN_dxyz[1, i] # dN_i/dy (shear strain in xy)
        B[3, i*3 + 1] = dN_dxyz[0, i] # dN_i/dx (shear strain in xy)

        B[4, i*3 + 1] = dN_dxyz[2, i] # dN_i/dz (shear strain in yz)
        B[4, i*3 + 2] = dN_dxyz[1, i] # dN_i/dy (shear strain in yz)

        B[5, i*3] = dN_dxyz[2, i] # dN_i/dz (shear strain in zx)
        B[5, i*3 + 2] = dN_dxyz[0, i] # dN_i/dx (shear strain in zx)

    return B
```

Element stiffness matrix $\mathbf{k}_e$

```
def element_stiffness_matrix(self):
    """
    Compute the stiffness matrix for the 8-node hexahedral element
    using numerical integration (Gauss quadrature)
    """
    Ke = np.zeros((24, 24)) # Initialize stiffness matrix
```

```
    # Loop over Gauss points
    for xi in self.gauss_pts:
        for eta in self.gauss_pts:
            for zeta in self.gauss_pts:

                # Compute shape function derivatives and Jacobian
                grad_N = self.shape_function_derivatives(xi, eta, zeta)
                J, invJ, detJ = self.jacobian_matrix(grad_N)

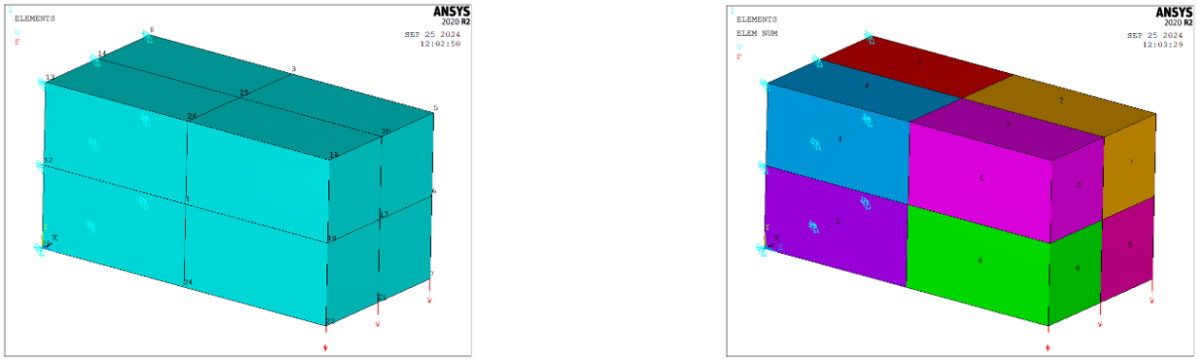
                # Compute B matrix and element stiffness contribution
                B = self.compute_B_matrix(invJ, grad_N)
                dK = B.T @ self.D @ B * detJ

                # Add to total stiffness matrix
                Ke += dK

    return Ke
```

.4.4 Comparison between ANSYS and in-house Implementation

To assess the differences between ANSYS and the in-house implementation of the strain-displacement matrix $\mathbf{B}$, specifically in terms of nodal strain and stress, a benchmark problem was analyzed. The test case consists of a cantilever beam subjected to a vertical force, modeled using four SOLID185 elements (see Fig. 10). The analysis was performed assuming linear, elastic, and isotropic material behavior.



(a) (b)

Figure 10: Isoparametric 8-node brick element: (a) node positions and numbering, (b) element positions and numbering.

The key features of the test problem are as follows:

- Geometry: Height $H = 1$ m, Width $B = 1$ m, Length $W = 2$ m;

- Material properties: $E = 5.4$ GPa, $\nu = 0.3$;
- Boundary conditions: fixed displacement at one end;
- Loading: $F = -1000$ N applied at the free end:
- Mesh: four hexahedral elements.

The focus is placed on element 1 (see Fig. 10b); for simplicity, calculations are shown only for this element and for the first Gauss point following ANSYS nodal ordering (see Table 4).

The nodal displacement vector for element 1, $\mathbf{u}_{e_1}$, is:

$$\mathbf{u}_{e_1} = \begin{bmatrix} 7.00807e-9 & -1.43608e-7 & -1.22085e-7 \\ 3.03178e-9 & -1.29821e-7 & 2.26395e-8 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1.00159e-21 & -1.49069e-7 & -1.21270e-7 \\ 6.66997e-22 & -1.34044e-7 & 2.15525e-8 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{48}$$

The element connectivity for element 1 is as following:

Node	x [m]	y [m]	z [m]
4	1.0	6e-17	1.0
2	1.0	0.5	1.0
9	1.0	0.5	0.0
10	1.0	0.0	0.0
23	0.5	6e-17	1.0
27	0.5	0.5	1.0
11	0.5	0.5	0.0
15	0.5	0.0	0.0

Table 6: Nodal coordinates of element 1.

The constitutive matrix is (see sec. .4.3):

$$\mathbf{D} = \begin{bmatrix} 7.27 \times 10^9 & 3.12 \times 10^9 & 3.12 \times 10^9 & 0 & 0 & 0 \\ 3.12 \times 10^9 & 7.27 \times 10^9 & 3.12 \times 10^9 & 0 & 0 & 0 \\ 3.12 \times 10^9 & 3.12 \times 10^9 & 7.27 \times 10^9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.08 \times 10^9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.08 \times 10^9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.08 \times 10^9 \end{bmatrix} \tag{49}$$

Being node 4 the first node in the nodal ordering, the associated Gauss point has the following coordinates:

- $r = -0.57735$
- $s = -0.57735$
- $t = -0.57735$

Consequently, the shape functions derivatives are (see sec.4.3):

$$\nabla \mathbf{N} = \begin{bmatrix} -8.33 \times 10^{-2} & 8.33 \times 10^{-2} & 3.11 \times 10^{-1} & -3.11 \times 10^{-1} & -2.23 \times 10^{-2} & 2.23 \times 10^{-2} & 8.33 \times 10^{-2} & -8.33 \times 10^{-2} \\ -3.11 \times 10^{-1} & -8.33 \times 10^{-2} & 8.33 \times 10^{-2} & 3.11 \times 10^{-1} & -8.33 \times 10^{-2} & 2.23 \times 10^{-2} & 2.23 \times 10^{-2} & 3.11 \times 10^{-1} \\ -8.33 \times 10^{-2} & -2.23 \times 10^{-2} & -8.33 \times 10^{-2} & -3.11 \times 10^{-1} & 8.33 \times 10^{-2} & 3.11 \times 10^{-1} & -2.23 \times 10^{-2} & 2.23 \times 10^{-2} \end{bmatrix} \tag{50}$$

The calculated Jacobian matrix $\mathbf{J}$, Jacobian determinant $\det \mathbf{J}$ and inverse of Jacobian matrix $\mathbf{J}^{-1}$ for node 4 (identified as node I in ANSYS node ordering) are (see sec.4.3):

$$\mathbf{J} = \begin{bmatrix} 0 & 0.25 & 0 \\ 0 & 0 & -0.5 \\ -0.25 & 0 & 0 \end{bmatrix} \tag{51}$$

$$\mathbf{J}^{-1} = \begin{bmatrix} 0 & 0 & -4 \\ 4 & 0 & 0 \\ 0 & -2 & 0 \end{bmatrix} \tag{52}$$

$$\det \mathbf{J} = 0.03125 \tag{53}$$

The associated $\mathbf{B}$ matrix is (see sec.4.3):

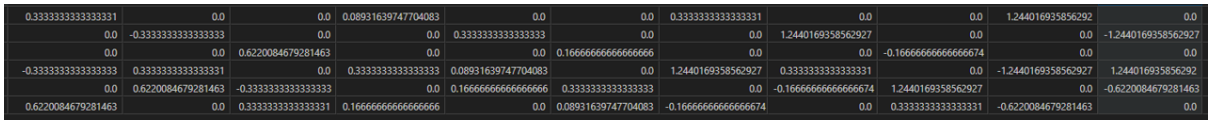


Figure 11: **B**: from 1^{st} to 11^{th} column.

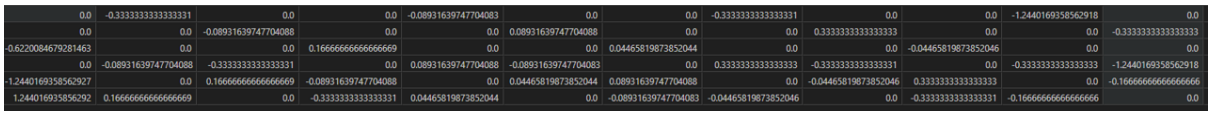


Figure 12: **B**: from 12^{th} to 23^{rd} column.

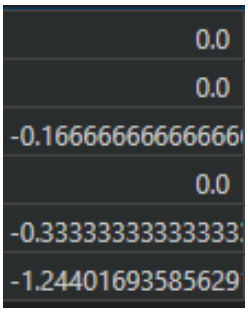


Figure 13: **B**: 24^{th} column.

To extract results at Gauss points, before solving the model it is important to insert the following command:

Listing 1: APDL script for results at intrgration points

```
ERESX,NO
```

To verify the correctness of the **B** matrix implementation, the deformation vector can be directly computed using Eq. 42.

Table 7 shows the strain components computed using ANSYS APDL and the in-house code for Element 1 at the first Gauss point. While the overall magnitudes are of the same order, noticeable discrepancies appear across several components. The normal strains ε_x and ε_z, as well as the shear strain τ_{yz}, show the most significant differences, indicating potential variations in the implementation of the **B** matrix or numerical integration. Further investigation is needed to identify the exact source of these deviations.

	ε_x	ε_y	ε_z	τ_{xy}	τ_{yz}	τ_{xz}
ANSYS APDL	1.52e-8	3.44e-8	-6.20e-8	2.34e-9	-4.02e-8	-7.70e-9
in-house code	9.73e-8	2.22e-8	-9.14e-8	3.25e-9	8.59e-8	4.21e-9

Table 7: Comparison of results from ANSYS APDL and the in-house code for rlement 1 at the first Gauss point.

.5 APDL implementation of the PSO algorithm

This appendix provides the ANSYS APDL script used to implement the Particle Swarm Optimization (PSO) algorithm for structural design optimization. The aim is to illustrate the integration of the optimization algorithm with the finite element model, highlighting how design variables, objective function evaluations, and swarm updates are handled within the APDL environment.

The script demonstrates the procedural setup, including particle initialization, fitness evaluation using ANSYS simulations, and the iterative process of updating positions and velocities according to the PSO rules. This code serves as a reference for those interested in applying gradient-free algorithms to structural analysis problems within ANSYS.

Recently, the PyMAPDL platform [26] has introduced support for directly coupling optimization algorithms, such as genetic algorithms from the PyGAD library, without requiring custom implementation within the APDL environment. This is particularly advantageous given the limited user-friendliness of APDL commands.

Listing 2: APDL PSO Script

```
! PSO algorithm
/title, PSO algorithm

FINISH
/CLEAR,ALL

/PREP7

! insert here the input (or data) which are not needed to repeat in the optimization (
    example: geometry, mesh, etc...)
/INPUT, your FE ".txt" file

! 0) FILE TO BE USED FOR OPTIMIZATION (to be filled):
*DIM,file_analysis,STRING,248 ! array used to store the string "file_analysis";
    file_analysis = file_path + file_name + file_extension
file_analysis(1,1) = your analysis file ! fill here

! 1) PROBLEM DEFINITION (to be filled):
n_DV = 2 ! number of design variables (DV)
! #lower boundaries:
lb = 10, 10 ! [mm]
! #upper boundaries:
ub = 50, 50 ! [mm]
! #constraints:
SVM_max_limit = 200 ! [MPa]
! #objective function reference (used to normalize the objective function and apply
```

```
    the penalty):
obj_fun_ref = 200*200*1000 ! [mm3]

! 2) PSO PARAMETERS (to be filled):
it_max = 10 ! max number of iteration
n_pop = 10 ! swarm size (or number of particles)
w = 1 ! inertia coefficient
w_damp = 0.9 ! damping coefficient
v_clamp = 0.1 ! velocity clamping factor
c1 = 2 ! cognitive coefficient (self acceleration); c1 = 2
c2 = 2 ! social coefficient (social acceleration); c2 = 2
!dt = 1 ! time step size (usually equal to 1) --> so it can be "excluded"


!!!*** START PSO ALGORITHM ***!!!
!!! From this line up to the end, do not change the code (only if you need to add some
     modifations on the algorithm) !!!
! #create array of bounds:
*DIM,bounds,ARRAY,2,n_DV
*DO,i,1,2
       *DO,j,1,n_DV
              *IF,i,EQ,1,THEN
                     bounds(i,j) = lb(j,1)
              *ELSE
                     bounds(i,j) = ub(j,1)
              *ENDIF
       *ENDDO
*ENDDO

! #create array of velocity limits:
*DIM,v_limits,ARRAY,2,n_DV
*DO,j,1,n_DV
       ! # v_max
       v_limits(2,j) = v_clamp*(bounds(2,j)-bounds(1,j)) ! max velocity for particle
           updating
       ! # v_min
       v_limits(1,j) = -v_limits(2,j) ! min velocity for particle updating
*ENDDO

! 3) INITIALIZATION:
! # parameters associated to each particle (which will be filled at each iteration):
! particle_pos    -->  particle position (array)   --> size = 1 x n_DV
! particle_vel    -->  particle velocity (array)      --> size = 1 x n_DV
! particle_obj_fun   -->  particle objective function (scalar)   --> size = 1 x 1
! best_particle_pos  -->  best particle position ever (array)   --> size = 1 x n_DV
! best_particle_obj_fun -->  best particle objective function ever (scalar) --> size =
     1 x 1
```

```
! best_global_obj_fun --> best global objective function in the swarm up to iteration
    k (scalar)
! particle = [particle_pos, particle_vel, particle_obj_fun, best_particle_pos,
    best_particle_obj_fun]
! DV = [DV1, DV2, ..., DVn] --> array used to transfer information between algorithm
    and solver --> size = 1 x n_DV (same as particle_pos)
! history_table = [DV1,DV2,...,DVn, obj_fun, SVM_max, d_max]

! # index used in the code:
! iter --> refers to i-th iterations
! i_n_pop --> referes to i-th population
! i_n_DV --> refers to i-th Design Variable

! N.B. To better understand the index used to get the different values into particle
    matrix, refers to the size of parameters above

! #create population matrix (referred as "particle"):
row = n_pop ! rows of particle matrix (equals to n of particle)
col = (n_DV*2)+1+n_DV+1 ! columns of particle matrix (equals to n of particle
    information to be saved at each iteration)
*DIM,particle,ARRAY,row,col ! particle matrix (which containts all the parameters
    above)

! #define Global Best Objective Function and Global Best Position (in the swarm)
    vectors:
*DIM,best_global_obj_fun_track,ARRAY,it_max+1,1 ! array to track global best obj.
    funct. estimated among the swarm at each iteration
*DIM,best_global_pos_track,ARRAY,it_max+1,n_DV ! array to track global best position
    estimated among the swarm at each iteration
*DIM,best_global_pos,ARRAY,1,n_DV ! best global position among the swarm up to
    iteration k
! #define position, velocity, best particle position and DV vectors:
*DIM,particle_pos,ARRAY,1,n_DV
*DIM,particle_vel,ARRAY,1,n_DV
*DIM,best_particle_pos,ARRAY,1,n_DV
*DIM,DV,ARRAY,1,n_DV
! #define history table:
*DIM,history_table,ARRAY,(it_max+1)*n_pop,n_DV+1+1 ! store results for each particle
    at each iteration

! #fill "best_global_obj_fun_track" with high values:
*DO,iter,1,it_max+1
       best_global_obj_fun_track(iter,1) = 2e64 ! modify this value if needed (must be
            high in case of minimization problem)
*ENDDO
! #initialize population members:
*DO,i_n_pop,1,n_pop ! where "i_n_pop" denotes the i-th particle in the population
```

```
*DO,i_n_DV,1,n_DV ! ! where "i_n_DV" denotes the i-th DV
       !#generate random initial solution:
       !particle_pos(1,i_n_DV) = RAND(bounds(1,i_n_DV),bounds(2,i_n_DV))
       !particle(i_n_pop,i_n_DV) = particle_pos(1,i_n_DV)
       !DV(1,i_n_DV) = particle_pos(1,i_n_DV)
       !#generate initial solution using LHS mehtod:
       range_DV = (bounds(2,i_n_DV)-bounds(1,i_n_DV))/n_pop ! i-th DV range
       *IF,i_n_pop,EQ,1,THEN
              particle_pos(1,i_n_DV) = RAND(bounds(1,i_n_DV) , bounds(1,i_n_DV
                  )+range_DV)
       *ELSEIF,i_n_pop,LT,n_pop,THEN
              particle_pos(1,i_n_DV) = RAND(bounds(1,i_n_DV)+range_DV*(i_n_pop
                  -1) , bounds(1,i_n_DV)+range_DV*i_n_pop)
       *ELSE
              particle_pos(1,i_n_DV) = RAND(bounds(1,i_n_DV)+range_DV*(i_n_pop
                  -1) , bounds(2,i_n_DV))
       *ENDIF
       particle(i_n_pop,i_n_DV) = particle_pos(1,i_n_DV)
       DV(1,i_n_DV) = particle_pos(1,i_n_DV)

       !#initialize velocity:
       particle_vel(1,i_n_DV) = 0
       particle(i_n_pop,n_DV+i_n_DV) = particle_vel(1,i_n_DV)

       !#fill history_table (add DVs):
       history_table(i_n_pop,i_n_DV) = particle_pos(1,i_n_DV)
*ENDDO

!#evaluation (after running FEA) of the objective function:
!##run FEA using the previous defined parameters:
/INPUT,file_analysis(1,1) ! analysis file
!##get obj. func. from FEA:
particle_obj_fun = obj_fun ! Objective Function

!#update the personal best of particle "i_n_pop" (that is the current position)
    :
*DO,i_n_DV,1,n_DV
       best_particle_pos(1,i_n_DV) = particle_pos(1,i_n_DV) ! at 1st iteration
           , best particle position is equal to initial particle position
*ENDDO
best_particle_obj_fun = particle_obj_fun ! at 1st iteration, best particle obj.
     funct. is equal to initial particle obj. funct.

!#fill particle matrix:
particle(i_n_pop,n_DV*2+1) = particle_obj_fun ! or "best_particle_obj_fun"
*DO,i_n_DV,1,n_DV ! put "best_particle_pos" in the right position of the matrix
       particle(i_n_pop,n_DV*2+1+i_n_DV) = best_particle_pos(1,i_n_DV)
```

```
        *ENDDO
        particle(i_n_pop,n_DV*2+1+n_DV+1) = best_particle_obj_fun

        !#update the "best_global_obj_fun_track" and "best_global_pos" vectors, getting
             the best obj. func. in the swarm:
        *IF,best_particle_obj_fun,LT,best_global_obj_fun_track(1,1),THEN
               best_global_obj_fun_track(1,1) = best_particle_obj_fun
               best_global_obj_fun = best_particle_obj_fun ! best objective function
                   found in the swarm
               *DO,i_n_DV,1,n_DV
                      best_global_pos(1,i_n_DV) = best_particle_pos(1,i_n_DV) ! get
                          the best global position associated with best global
                          objective function in the swarm
                      best_global_pos_track(1,i_n_DV) = best_particle_pos(1,i_n_DV)
               *ENDDO
        *ENDIF

        !#fill history_table (add obj. func. and max stress):
        history_table(i_n_pop,n_DV+1) = particle_obj_fun ! obj. func.
        history_table(i_n_pop,n_DV+2) = svm_max ! max stress
*ENDDO

! 4) PSO MAIN LOOP:
global_index = 1 ! index used to track all solutions calculated by the PSO
*DO,iter,2,it_max+1 ! iter = i-th iteration --> from 2 to the n of iteration (the
    outer loop start from 2 because of the particle inizialization)
       *DO,i_n_pop,1,n_pop ! i_n_pop = i-th particle --> from 1 to the n of particle
           in the swarm

              !#define DV(s):
              *DO,i_n_DV,1,n_DV ! i_n_DV = i_n_DV-th DV --> from 1 to the n of Design
                   Variables
                     !#calculate particle velocity and particle position:
                     r1 = RAND(0,1)
                     r2 = RAND(0,1)
                     particle_vel(1,i_n_DV) = w*particle(i_n_pop,n_DV+i_n_DV) + c1*r1
                         *(particle(i_n_pop,n_DV*2+1+i_n_DV)-particle(i_n_pop,i_n_DV)
                         ) + c2*r2*(best_global_pos(1,i_n_DV)-particle(i_n_pop,i_n_DV
                         ))
                     particle_pos(1,i_n_DV) = particle(i_n_pop,i_n_DV) + particle_vel
                         (1,i_n_DV)

                     !#check if particle violates bounds (and recalculate velocity
                         and position if needed):
                     *IF,particle_pos(1,i_n_DV),GT,bounds(2,i_n_DV),OR,particle_pos
                         (1,i_n_DV),LT,bounds(1,i_n_DV),THEN
                            *IF,particle_vel(1,i_n_DV),LT,v_limits(1,i_n_DV),THEN ! v
```

```
            < v_min
               particle_vel(1,i_n_DV) = v_limits(1,i_n_DV)
         *ELSEIF,particle_vel(1,i_n_DV),GT,v_limits(2,i_n_DV),THEN
            ! v > v_max
               particle_vel(1,i_n_DV) = v_limits(2,i_n_DV)
         *ELSE ! v_min <= v <= v_max --> keep the velocity
             calculated so far
               particle_vel(1,i_n_DV) = particle_vel(1,i_n_DV) !
                   do not change particle_vel
         *ENDIF
         !#recalculate particle position:
         particle_pos(1,i_n_DV) = particle(i_n_pop,i_n_DV) +
             particle_vel(1,i_n_DV)
         !#recheck if bounds are violated:
         !##mirror effect:
         *IF,particle_pos(1,i_n_DV),GT,bounds(2,i_n_DV),OR,
             particle_pos(1,i_n_DV),LT,bounds(1,i_n_DV),THEN
               particle_vel(1,i_n_DV) = -w*particle(i_n_pop,n_DV+
                   i_n_DV) + c1*r1*(particle(i_n_pop,n_DV*2+1+
                   i_n_DV)-particle(i_n_pop,i_n_DV)) + c2*r2*(
                   best_global_pos(1,i_n_DV)-particle(i_n_pop,
                   i_n_DV))
               particle_pos(1,i_n_DV) = particle(i_n_pop,i_n_DV)
                   + particle_vel(1,i_n_DV)
               !##recheck for last time if bounds are violated:
               *IF,particle_pos(1,i_n_DV),GT,bounds(2,i_n_DV),
                   THEN ! particle_pos > ub
                        particle_pos(1,i_n_DV) = bounds(2,i_n_DV)
                            ! set particle position to upper bound
                             --> particle_pos = ub
               *ELSEIF,particle_pos(1,i_n_DV),LT,bounds(1,i_n_DV)
                   ,THEN ! particle_pos < lb
                      particle_pos(1,i_n_DV) = bounds(1,i_n_DV) !
                           set particle position to lower bound
                          --> particle_pos = lb
               *ENDIF
         *ENDIF
      *ENDIF


      !#update particle matrix and DV with current position and
          velocity:
      particle(i_n_pop,i_n_DV) = particle_pos(1,i_n_DV)
      particle(i_n_pop,i_n_DV+n_DV) = particle_vel(1,i_n_DV)
      DV(1,i_n_DV) = particle_pos(1,i_n_DV)

      !#fill history_table (add DVs):
```

```
       history_table((n_pop+global_index),i_n_DV) = particle_pos(1,
           i_n_DV)
*ENDDO


!#evaluation (after running FEA) of the objective function:
/INPUT,file_analysis(1,1) ! analysis file
particle_obj_fun = obj_fun ! Objective Function

!#update current particle objective function:
particle(i_n_pop,n_DV*2+1) = particle_obj_fun

!#update the personal best position and objective function of particle
    "i_n_pop":
*IF,particle_obj_fun,LT,particle(i_n_pop,n_DV*2+1+n_DV+1),THEN
       !#obj. fun. updating:
       best_particle_obj_fun = particle_obj_fun
       particle(i_n_pop,n_DV*2+1+n_DV+1) = best_particle_obj_fun
       !#position updating:
       *DO,i_n_DV,1,n_DV
              particle(i_n_pop,n_DV*2+1+i_n_DV) = particle_pos(1,i_n_DV
                  )
       *ENDDO
*ENDIF

!#update best obj. fun. found at each iteration:
*IF,i_n_pop,EQ,1,THEN ! 1st particle (or solution)
       !best_global_obj_fun_track(iter,1) = particle_obj_fun
       *DO,i_n_DV,1,n_DV
              best_global_pos_track(iter,i_n_DV) = particle_pos(1,
                  i_n_DV)
       *ENDDO
*ELSEIF,best_particle_obj_fun,LT,best_global_obj_fun_track(iter,1),THEN
     ! other particles (or solutions)
       best_global_obj_fun_track(iter,1) = particle_obj_fun
       *DO,i_n_DV,1,n_DV
              best_global_pos_track(iter,i_n_DV) = particle_pos(1,
                  i_n_DV)
       *ENDDO
*ENDIF

!#update the global best objective function and position ever reached
    in the swarm at current iteration:
*IF,best_particle_obj_fun,LT,best_global_obj_fun,THEN
       best_global_obj_fun = best_particle_obj_fun
       best_global_obj_fun_track(iter,1) = best_particle_obj_fun ! if
           best particle is better, update best global obj. fun.
```

```
                *DO,i_n_DV,1,n_DV
                    best_global_pos(1,i_n_DV) = particle_pos(1,i_n_DV)
                *ENDDO
            *ELSE
                best_global_obj_fun_track(iter,1) = best_global_obj_fun_track(
                    iter-1,1) ! if best particle is not better, keep best global
                     obj. fun.
            *ENDIF
            !#fill history_table (add obj. func. and max stress):
            history_table(n_pop+global_index,n_DV+1) = particle_obj_fun ! obj. func
                .
            history_table(n_pop+global_index,n_DV+2) = svm_max ! max stress

            global_index = global_index + 1
        *ENDDO
        !#update inertia coefficient:
        w = w*w_damp

        ! save results at each loop:
        PARSAV,ALL,"results_checkpoint","txt"
*ENDDO

! 5) PLOT RESULTS OF OPTIMIZATION:
*DIM,results_opt,TABLE,(it_max+1),2 ! table used to store normalized obj. fun. and n
    of iterations
*DO,iter,1,(it_max+1)
        results_opt(iter,1) = iter ! i-th iteration
        results_opt(iter,2) = best_global_obj_fun_track(iter,1) ! i-th best global
            objective function
*ENDDO
/GTHK,AXIS,3 ! sets line thicknesses for graph lines --> axis
/GTHK,GRID,1 ! sets line thicknesses for graph lines --> grid
/AXLAB,X,Number of Iterations ! x label
/AXLAB,Y,Normalized objective function ! y label
*VPLOT,results_opt(1,1),results_opt(1,2)

! save results in the database (in the current working directory):
SAVE,results_of_optimization,DB

! write into a txt file all the parameters and arrays in the file (in the current
    working directory):
PARSAV,ALL,'opt_par_and_array','txt'

!!!*** END PSO ALGORITHM ***!!!
```