Scuola IMT Alti Studi Lucca

Constraint-driven optimization and parametrization of industrial NURBS geometries via neural deformation field

Questa è la versione preprint della seguente opera:

Original
Constraint-driven optimization and parametrization of industrial NURBS geometries via neural deformation field / Tamburlin, F., Canali, G., D'Inverno, G.A., Demo, N., Mola, A., Rozza, G.. - In: COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING. - ISSN 0045-7825. - 464:(2027). [10.1016/j.cma.2026.119500]

Availability:
This version is available at: 20.500.11771/45440

Publisher:
Elsevier

Published
DOI:10.1016/j.cma.2026.119500

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

23 November 2026

Highlights

Constraint-driven Optimization and Parametrization of Industrial NURBS Geometries via Neural Deformation Field

Federico Tamburlin, Giovanni Canali, Giuseppe Alessio D'Inverno, Nicola Demo, Andrea Mola, Gianluigi Rozza

- Differentiable CAD-native pipeline using neural parametrizations to optimize NURBS geometry via integral constraints.
- End-to-end gradient propagation from loss to control points — no adjoint equations needed.
- CAD-compatible output (IGES) preserving smoothness and topology for industrial downstream processes.

arXiv:2606.07198v1 [math.NA] 5 Jun 2026

Constraint-driven Optimization and Parametrization of Industrial NURBS Geometries via Neural Deformation Field

Federico Tamburlin[a], Giovanni Canali[a], Giuseppe Alessio D'Inverno[a], Nicola Demo[a,c], Andrea Mola[b], Gianluigi Rozza[a,c]

[a] *SISSA, International School for Advanced Studies, Trieste, 34136, Italy*
[b] *IMT School for Advanced Studies Lucca, Lucca, 55100, Italy*
[c] *FAST Computing Srl, Trieste, 34136, Italy*

Abstract

This work presents a differentiable framework for the parametrization and shape optimization of industrial CAD geometries represented by multi-patch NURBS surfaces. The method enables the deformation of complex CAD models through a physics-informed geometric parametrization, allowing direct morphing driven by physical constraints without the need to prescribe a predefined deformation strategy. A neural displacement field, implemented as a multi-layer perceptron acting on the NURBS control points, provides a compact parametrization of the admissible design space while preserving patch connectivity.

Global geometric quantities relevant to hydrostatic design, including displaced volume, wetted surface area and buoyancy centroid, are formulated as differentiable integral operators evaluated on the parametric domain. These quantities are computed through Gauss–Legendre quadrature combined with analytical B-spline derivatives for surface metric evaluation, allowing gradient propagation to the deformation parameters while limiting the computational overhead of automatic differentiation.

The proposed framework operates directly on CAD representations without intermediate mesh generation. Numerical experiments on a modified KVLCC2 hull demonstrate the capability of the method to satisfy competing hydrostatic constraints while producing smooth CAD-compatible geometries and showing stable convergence across multiple random initializations.

Keywords: Shape optimization, NURBS, Physics-Informed Neural Networks, Computer-Aided Design, Neural parameterization
2020 MSC: 65D17, 68T07, 49Q10, 65K10

1. Introduction

1.1. Motivation

Computer-Aided Design (CAD) models are the native representation of industrial geometries in Product Lifecycle Management (PLM) [1, 2] and Computer-Aided Manufacturing (CAM) [3]. Free-form shapes are commonly described through Boundary Representation (B-Rep) models based on Non-Uniform Rational B-Splines (NURBS) [4], which provide an exact parametrization of curves and surfaces through control points, weights, knot vectors, and polynomial degrees. Preserving the geometric and topological validity of this representation during deformation is essential for downstream analysis and manufacturing.

Traditional shape optimisation workflows [5–7] require the definition of a reduced set of design parameters capable of producing meaningful geometric variations while preserving the validity of the underlying shape representation. Classical deformation-based parametrizations, such as Free-Form Deformation (FFD) [8] and Radial Basis Function (RBF) morphing, are widely used in these pipelines because they can modify complex geometries without rebuilding the CAD model from scratch. They can act on different representations, including CAD control entities, surface discretizations, point clouds, and volumetric meshes, and have been successfully applied in industrial and naval shape optimization through geometric, mesh, and adaptive grid deformation strategies [7, 9–11]. However, their effectiveness strongly depends on the definition of the deformation structure, such as the FFD lattice, the RBF control points, the

support radius, or the propagation strategy. Coarse deformation models may fail to capture localized features, whereas finer parametrizations increase the number of design variables and may introduce spurious high-frequency modes [12]. Moreover, since these deformation maps are usually external to the native CAD parametrization, preserving geometric regularity, derivative information, and global CAD-level constraints is not automatic, especially when the quantities of interest are integral properties rather than pointwise displacements or mesh-based measures.

In naval hydrodynamics, several works have addressed hull-form optimization through advanced deformation strategies acting either on the geometry or directly on the computational mesh. [9] proposed a three-dimensional deformation strategy for hull-form optimization, while [7] introduced an effective mesh deformation approach for hull shape design optimization based on CAD-consistent geometric modifications and volumetric mesh propagation. Similarly, a combined mesh deformation with adaptive refinement in hull-form design was introduced in [10], and [11] presented a CFD-based optimization framework in calm water using adaptive grid deformation. These contributions demonstrate the practical relevance of deformation-based parametrizations in naval applications.

Alongside classical mesh and geometry deformation techniques, data-driven and generative approaches have recently emerged as promising tools to construct compact design spaces for complex industrial shapes. In [13] authors argued that traditional CAD parameters may limit the exploration of large shape variations and topological changes, whereas generative networks can provide low-dimensional manifolds for design optimization. In the naval field, [14] proposed a deep convolutional generative model for parametric hull design, showing how generative networks can learn compact representations of admissible ship hull shapes. [15] introduced a diffusion-based model for parametric ship hull generation under multiple objectives and constraints, whereas more generally, [16] focused on generative models for the deformation of industrial shapes subject to linear geometric constraints, with emphasis on model-order and parameter-space reductions. These approaches highlight a growing interest in replacing high-dimensional, manually defined shape parameters with reduced latent or generative representations. However, implicit and generative descriptions may reduce direct control over standard CAD entities and may require additional mechanisms to guarantee strict geometric validity.

The advent of Deep Learning and Physics-Informed Neural Networks has also promoted the use of Automatic Differentiation as a mechanism for differential equation solution [17]. This makes it possible to compute sensitivities without deriving problem-specific adjoint codes [18], which are traditionally required in conventional gradient-based shape optimization workflows [19, 20]. Differentiable programming further allows geometric kernels to be embedded directly into gradient-based loops. [21] proposed *THB-Diff*, exploiting Automatic Differentiation on GPUs for Truncated Hierarchical B-splines fitting and refinement, mainly targeting computational efficiency in spline evaluation. Current PINN-based shape optimization frameworks [22, 23] often rely on coordinate projection strategies or problem-specific parametrizations, differing from the direct optimization of standard CAD control variables.

In this context, the present work follows a complementary direction. Rather than relying exclusively on mesh-dependent deformation pipelines [7, 10, 11], stochastic or generative design manifolds [14–16], or soft-constrained PINN formulations [22, 23], it defines an analytical and differentiable map from NURBS control points to global geometric metrics. Automatic Differentiation is combined with numerical quadrature to enforce integral geometric properties directly at the CAD-parametrization level, rather than purely minimizing fitting errors or evaluating constraints on a generated mesh. This enables gradient-based optimization while preserving explicit control over industrial shape deformation and bypassing explicit mesh generation during the constraint-evaluation stage.

1.2. Main novelties and contributions

This work presents a shape design and optimization pipeline for parametric surfaces (Figure 1), implemented within the PINA framework [24]. The proposed approach focuses on a simplified physical setting, where the quantities of interest are global geometric and hydrostatic measures, such as volume, surface area, and centroid-related properties, evaluated through analytical B-spline derivatives and numerical quadrature. While full PDE-constrained shape optimization remains the standard approach for many engineering applications, its integration within a differentiable CAD-native framework introduces significant complexity and is out of the scope of the present work, representing a natural direction for future extensions. Instead, this paper introduces a constraint-driven deformation paradigm, where the geometry is optimized with respect to physically meaningful integral targets, avoiding explicit field equation solves while retaining a clear physical interpretation. In this context, the framework can be interpreted within the broader class of PINNs, where physical consistency is enforced through differentiable constraints embedded in the loss function, and not with data. The deformation is modeled via a Neural Displacement Field, implemented as a Multi-Layer Perceptron

(MLP), which maps the original CAD control points to their deformed configuration, enabling a compact and flexible parametrization of the design space.

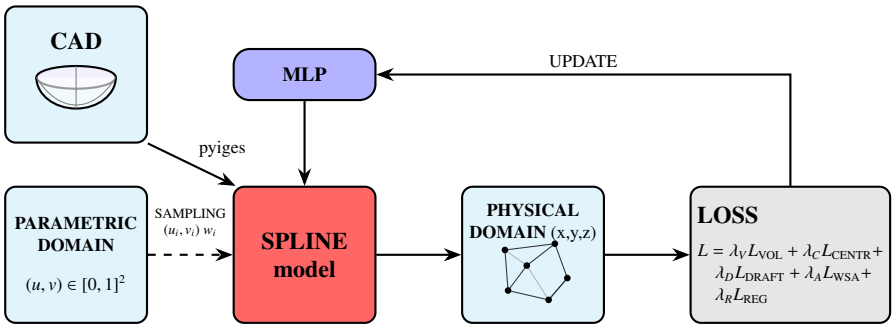


Figure 1: Schematic of the end-to-end differentiable optimization pipeline. The system maps original geometrical control points through a Neural Displacement Field (MLP) to generate the deformed surface. Geometric properties (Volume, Area, Centroid) are computed via analytical B-Spline derivatives and numerical quadrature, driving the loss function composed of integral constraints, geometric barrier and regularization terms.

The main contributions of this paper are summarized as follows:

- **Neural Parametrization of Geometric Deformation:** An approach employing a Multi-layer Perceptron (MLP) to parametrize the displacement field of NURBS control points. This provides a flexible and compact representation of the design space compared to point-by-point optimization and establishes a global deformation law that maintains interface continuity [25, 26].
- **End-to-End Differentiable Mapping:** By integrating the B-Spline mapping directly into the computational graph, the framework enables exact propagation of gradients from the objective function to the control points. This eliminates the need for manually derived adjoint equations.
- **Enforcement of Integral Geometric Constraints:** The formulation of differentiable loss terms to satisfy global integral constraints, such as target volume. This ensures that the optimized geometry adheres to functional global requirements, implemented within the PINA framework.
- **CAD Compatibility:** The pipeline operates directly on standard NURBS patch parameters, maintaining the topological validity and smoothness required for industrial downstream processes. The output is a standard CAD file (e.g., IGES).

1.3. Outline of the paper

The remainder of this paper is organized as follows. Section 2 introduces the main theoretical concepts and notation. Section 3 details the implementation within the PINA framework and the neural parametrization of deformation. Section 4 presents numerical experiments validating the proposed approach on a multi-patch ship geometry. Finally, Section 5 summarizes the findings and outlines future possible research directions.

2. Preliminary notation and concepts

This section outlines the mathematical formulation of the geometric representation and the numerical integration scheme adopted within the differentiable optimization pipeline.

2.1. B-Splines Surface Parametrization

Let $\hat{\Omega} = [0, 1]^2 \subset \mathbb{R}^2$ denote the parametric reference domain. A physical surface $\Omega \subset \mathbb{R}^3$ is considered, defined by the mapping $\mathbf{S} : \hat{\Omega} \to \Omega$. In the context of standard CAD formats, the geometry is represented as a Non-Uniform Rational B-Spline (NURBS) surface, which is a subset of the B-Spline employed in this work:

$$\mathbf{S}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}^{p,q}(u, v)\, \mathbf{C}_{i,j}, \quad (u, v) \in \hat{\Omega}, \tag{1}$$

where $\mathbf{C}_{i,j} \in \mathbb{R}^3$ are the control points forming the control net. The rational basis functions $R_{i,j}^{p,q}$ are defined as:

$$R_{i,j}^{p,q}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)w_{k,l}}, \tag{2}$$

where $N_{i,p}(u)$ and $N_{j,q}(v)$ are the univariate B-spline basis functions of degree p and q, defined over the knot vectors $\Xi_u = \{\xi_0, \ldots, \xi_{n+p+1}\}$ and $\Xi_v = \{\eta_0, \ldots, \eta_{m+q+1}\}$, respectively, and $w_{i,j}$ are the associated weights [27].

In the proposed framework, the topological parameters, specifically the knot vectors, polynomial degrees, and weights, are fixed to the initial geometry values. The spatial coordinates of the control points $\mathbf{C} = \{\mathbf{C}_{i,j}\}$ constitute the set of design variables subject to optimization.

2.2. Geometry and Metric Distortion

Evaluation of geometric quantities on the physical manifold Ω requires the computation of the metric distortion induced by the mapping $\mathbf{S}$. The tangent basis vectors are given by the partial derivatives $\partial_u\mathbf{S}$ and $\partial_v\mathbf{S}$. The local surface area element $d\Omega$ is related to the parametric area element $d\hat{\Omega}$ via the Jacobian determinant J, defined as the magnitude of the normal vector field:

$$J(u,v) = \|\mathbf{n}(u,v)\|_2, \quad \text{with} \quad \mathbf{n}(u,v) = \frac{\partial\mathbf{S}}{\partial u} \times \frac{\partial\mathbf{S}}{\partial v}. \tag{3}$$

This Jacobian term ensures that gradients computed in the parametric space correctly reflect the sensitivity of physical objective functionals in the Euclidean space. The computation of the tangent vectors relies on the exact analytical derivatives of the univariate B-spline basis functions. For a basis function $N_{i,p}(u)$ of degree p defined on a knot vector U, the derivative is calculated recursively as:

$$\frac{d}{du}N_{i,p}(u) = \frac{p}{u_{i+p} - u_i}N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}}N_{i+1,p-1}(u). \tag{4}$$

2.3. Integral Constraints and Numerical Quadrature

Global geometric properties, such as volume and centroids, are evaluated via numerical integration over the parametric domain. A Gauss-Legendre [28] quadrature rule is employed with N_q nodes $\{(\hat{u}_k, \hat{v}_k)\}_{k=1}^{N_q}$ and corresponding weights $\{\omega_k\}_{k=1}^{N_q}$. The approximation of a generic scalar field integral $I = \int_\Omega f(\mathbf{x})\, d\Omega$ is given by:

$$I \approx \sum_{k=1}^{N_q} f(\mathbf{S}(\hat{u}_k, \hat{v}_k))\, J(\hat{u}_k, \hat{v}_k)\, \omega_k. \tag{5}$$

2.3.1. Volumetric and Surface Constraints

For hydrodynamic applications, the geometry is often modelled as an open surface bounded by a reference plane at $z = z_{\text{ref}}$. The enclosed volume V is computed by integrating the vertical elevation weighted by the projected area on the xy-plane. Leveraging the divergence theorem, this is formulated as:

$$V = \int_\Omega (z - z_{\text{ref}})\, n_z\, d\Omega \approx \sum_{k=1}^{N_q} (S_z(\hat{u}_k, \hat{v}_k) - z_{\text{ref}})\, n_z(\hat{u}_k, \hat{v}_k)\, J(\hat{u}_k, \hat{v}_k)\, \omega_k, \tag{6}$$

where n_z is the z-component of the unit normal. In the numerical implementation, the term $n_z J$ is computed efficiently as the z-component of the un-normalized cross product $\partial_u\mathbf{S} \times \partial_v\mathbf{S}$, avoiding explicit normalization steps.

The Center of Buoyancy (COB) is evaluated directly on the displaced volume by computing the static moments of the submerged geometry. Leveraging the divergence theorem on the bounding surface, the spatial coordinates are formulated as:

$$\mathbf{x}_{COB} = \frac{\int_\Omega \mathbf{X}(z - z_{ref})n_z\, d\Omega}{\int_\Omega (z - z_{ref})n_z\, d\Omega}, \quad \mathbf{X} = \begin{bmatrix} x \\ y \\ \frac{z+z_{ref}}{2} \end{bmatrix}. \tag{7}$$

where the term $\frac{z+z_{\text{ref}}}{2}$ represents the vertical lever arm of the elementary volume column.

Finally, the Wetted Surface Area (WSA) corresponds to the surface area of the manifold:

$$A = \int_{\Omega} \mathbb{1}_{z \le z_{\text{ref}}} \, d\Omega \approx \sum_{k=1}^{N_q} \mathbb{1}_{z(\hat{u}_k, \hat{v}_k) \le z_{\text{ref}}} \, J(\hat{u}_k, \hat{v}_k) \, \omega_k. \tag{8}$$

The evaluation of these quantities and their differentiability is central to the proposed optimization pipeline.

3. Methodological Framework

Consider a computational domain $\Omega \subset \mathbb{R}^3$ defined by a set of NURBS patches, in the physical space. The shape optimization problem is formulated as the search for a deformation field that satisfies global integral and geometrical constraints while minimizing a regularity functional. The proposed framework relies on a global neural parametrization of the displacement field, integrated into a differentiable computing environment.

3.1. Neural Deformation Operator

Let $C^0 = \{\mathbf{P}_i^0\}_{i \in \mathcal{I}}$ be the set of control points defining the reference geometry Ω_0. Let us define a continuous deformation map $\Phi_{\boldsymbol{\theta}} : \mathbb{R}^3 \to \mathbb{R}^3$, parametrized by a Multi-Layer Perceptron (MLP) with trainable weights $\boldsymbol{\theta}$ (Figure 2). To ensure a balance between expressivity and geometric regularity, the architecture comprises 5 hidden layers with hyperbolic tangent activation, followed by a linear projection to the physical space $\mathbb{R}^3$. This depth was selected based on preliminary benchmarks: deeper networks (> 8 layers) tended to introduce optimization instability and unwanted high-frequency artifacts, reducing the implicit regularization effect. Conversely, shallower networks (< 3 layers) lacked the spectral capacity to capture localized shape variations, such as the deformation required at the bulbous bow. The set of deformed control points $C(\boldsymbol{\theta}) = \{\mathbf{P}_i(\boldsymbol{\theta})\}_{i \in \mathcal{I}}$ is defined by applying the displacement field to the reference set C^0:

$$\mathbf{P}_i(\boldsymbol{\theta}) = \mathbf{P}_i^0 + \Phi_{\boldsymbol{\theta}}(\mathbf{P}_i^0), \quad \forall i \in \mathcal{I}. \tag{9}$$

Since $\Phi_{\boldsymbol{\theta}}$ is a deterministic smooth function (C^∞), topological connectivity at patch interfaces is satisfied by construction. For any shared boundary $\Gamma_{AB} = \Omega_A \cap \Omega_B$, coincident control points receive identical displacement vectors, preserving the water-tightness of the assembly. To enforce symmetry with respect to the plane $y = 0$, the operator output is explicitly constrained. The hard constraint is imposed:

$$\Phi_{\boldsymbol{\theta},y}(\mathbf{x}) = 0, \quad \forall \mathbf{x} \text{ s.t. } |y| < \epsilon. \tag{10}$$

Analogously, a hard constraint is imposed on the vertical displacement of the upper boundary. Letting z_{top} denote the maximum vertical coordinate of the reference hull, the operator is constrained such that:

$$\Phi_{\boldsymbol{\theta},z}(\mathbf{x}) = 0, \quad \forall \mathbf{x} \text{ s.t. } |z - z_{top}| < \epsilon. \tag{11}$$

The network $\Phi_{\boldsymbol{\theta}}$ is initialized with a distribution ensuring zero initial displacements, allowing the optimization to start from the exact reference configuration Ω_0.

3.2. Integral Constraints and Geometric Measures

The shape optimization objectives are formulated as global integral quantities computed over the parametric domain. The PINA framework is extended to evaluate global geometric integrals via numerical quadrature. A hybrid differentiation scheme is implemented: the neural network parameters are optimized via standard automatic differentiation, while the surface geometric properties (normals and Jacobian determinants) are computed using exact analytical B-Spline basis derivatives. This avoids the explicit automatic differentiation of the B-spline kernel for tangent vector computation, reducing the computational graph overhead by approximately 50% [1].

[1] Benchmarks on the Jacobian evaluation per epoch show a reduction from 0.365s (Automatic Differentiation) to 0.193s (Analytical PINA derivative), confirming a computational saving of $\approx 47\%$

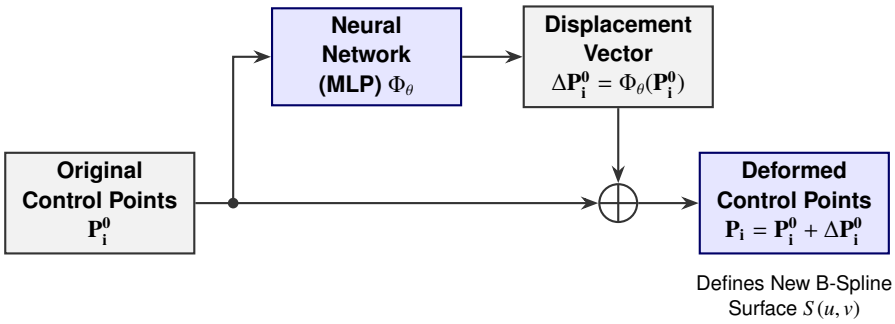


Figure 2: Neural Deformation architecture: the displacement is computed by the MLP and added to the original control points.

Let $Q(\Omega)$ be a generic quantity of interest defined as a rational integral:

$$Q(\Omega) = \frac{\int_{\hat{\Omega}} N(\mathbf{u}(\boldsymbol{\xi}))J(\boldsymbol{\xi})d\boldsymbol{\xi}}{\int_{\hat{\Omega}} D(\mathbf{u}(\boldsymbol{\xi}))J(\boldsymbol{\xi})d\boldsymbol{\xi}} \approx \frac{\sum_k N_k J_k w_k}{\sum_k D_k J_k w_k}, \tag{12}$$

where $\mathbf{u}$ is the displacement field, and J_k is the geometric measure derived from the analytical B-Spline derivatives. Specifically, $J = \|\mathbf{n}\|$ is used for surface metrics and $J = \mathbf{n} \cdot \mathbf{e}_z$ for volumetric quantities via the Divergence Theorem. The specific constraints are listed in Table 1.

Table 1: Definition of Integral Constraints formulation. Operators N and D refer to Equation (12). For entries marked with $\emptyset$, the denominator is set to unity (scalar integral). Note: $\mathbf{X}$ is the vector $[x, y, \frac{z+z_{ref}}{2}]$.

Constraint	$N(\mathbf{x})$	$D(\mathbf{x})$	Measure J	Notes
Displaced Volume	$z - Z_{ref}$	$\emptyset$	$\mathbf{n} \cdot \mathbf{e}_z$	Projection on xy-plane
Wetted Surface	$\mathbb{1}_{z \le z_{ref}}$	$\emptyset$	$\|\mathbf{n}\|$	Submerged area ($z \le z_{ref}$)
Center of Buoyancy	$\mathbf{X}(z - z_{ref})$	$z - z_{ref}$	$\mathbf{n} \cdot \mathbf{e}_z$	Volume moments

3.3. Optimization Problem Statement

The optimization is formulated as an unconstrained minimization problem of a composite scalar loss function $\mathcal{L}(\boldsymbol{\theta})$.

Problem 1 (Shape Optimization). *Find the optimal network parameters $\boldsymbol{\theta}^* \in \mathbb{R}^d$ such that:*

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\mathcal{L}_{int}(\boldsymbol{\theta}) + \mathcal{L}_{barrier}(\boldsymbol{\theta}) + \mathcal{L}_{reg}(\boldsymbol{\theta}) \right), \tag{13}$$

subject to the hard constraints on symmetry and vertical displacement bounds.

The *Integral Constraints* $\mathcal{L}_{int}$, computed as the deviation from target geometric properties, (Volume V, Area A and Center of Buoyancy x_{COB}) are minimized via a Sum of Relative Squared Errors:

$$\mathcal{L}_{int} = \lambda_V \left(\frac{V(\boldsymbol{\theta}) - V_{targ}}{V_{targ}} \right)^2 + \lambda_A \left(\frac{A(\boldsymbol{\theta}) - A_{targ}}{A_{targ}} \right)^2 + \lambda_C \left(\frac{x_{COB}(\boldsymbol{\theta}) - x_{COB,targ}}{x_{COB,targ}} \right)^2. \tag{14}$$

A specific component mask is applied to the COB condition to penalize exclusively the longitudinal shift (x_{COB}), allowing the transverse and vertical components (y_{COB}, z_{COB}) to naturally adapt to the new displaced mass distribution.

The *Geometric Barrier* $\mathcal{L}_{barrier}$ constraint, a unilateral constraint on the draft (minimum vertical coordinate Z_{min}), is enforced via a penalty method. The violation set is defined, based on the Rectified Linear Unit (ReLU) operator:

$$\mathcal{L}_{barrier} = \lambda_D \sum_{k=1}^{N_q} [\max(0, Z_{min} - z(\mathbf{x}_k(\boldsymbol{\theta})))]^2 . \tag{15}$$

A displacement-dependent energy functional is introduced, the *Regularization Functional* $\mathcal{L}_{reg}$, to ensure surface quality and suppress high-frequency geometric artifacts. Unlike Euclidean regularization on parameters, this term acts on the physical displacement $\mathbf{u}(\boldsymbol{\xi})$ numerically integrated over the manifold:

$$\mathcal{L}_{reg} = \lambda_{reg} \sum_{k=1}^{N_q} \left[\exp\left(\gamma_{reg} \|\mathbf{u}(\boldsymbol{\xi}_k)\|^2 \right) - 1 \right] J(\boldsymbol{\xi}_k) w_k. \tag{16}$$

The exponential kernel imposes an asymptotic penalty as displacements approach the soft limit defined by $\gamma_{reg}^{-1/2}$, implicitly constraining the search space to small deformations, under a certain tunable threshold.

3.4. Optimization Scheme

The continuous problem is discretized via Gaussian quadrature [28]. The discrete optimization procedure is detailed in Algorithm 1. The iterative scheme is structured into four primary stages. First, the forward pass of the neural network computes the unconstrained displacement field $\Delta\tilde{\mathcal{C}}$, which is subsequently projected onto the feasible space via the operator $\mathcal{P}$ to strictly enforce symmetry and boundary conditions.

Second, to minimize computational overhead, a geometric caching strategy is implemented. The physical coordinates $\mathbf{x}_q$ and the metric vectors $\mathbf{m}_q$ are evaluated exclusively at the invariant quadrature nodes. This approach isolates the exact evaluation of the B-spline analytical derivatives from the automatic differentiation graph associated with the neural network parameters.

Third, the functional assembly aggregates the geometric residuals $\mathcal{R}$ and the regularization energy $\mathcal{E}_{reg}$ via numerical integration. Finally, the network weights $\boldsymbol{\theta}$ are updated. The abstract operator Optimizer relies on the exact gradients $\nabla_\theta \mathcal{L}$ extracted via automatic differentiation, utilizing AdamW for initial non-linear exploration, followed by the L-BFGS algorithm for second-order local refinement.

4. Results

4.1. Integral convergence analysis

The accurate evaluation of global geometric constraints necessitates a numerical integration scheme. A Gauss-Legendre [28] quadrature rule is applied over the parametric domain $\hat{\Omega} = [0, 1]^2$ to approximate surface and volumetric integrals.

As illustrated in Figure 3, the volumetric computation exhibits convergence behaviour. Despite these initial error, the scheme achieves stabilisation.

For the optimization framework, a resolution of $N_q = 18$ points per dimension was established as a trade-off.

4.2. Experimental Setup and Initial Benchmark

The proposed framework was validated on a modified version of the *KVLCC2* hull geometry[2] [29]. While the framework is extensible to trimmed topologies, this experimental setup leverages the native tensor-product structure of untrimmed patches. The optimization pipeline was executed on a consumer-grade mobile workstation[3].

The optimization is performed using a fixed random seed to ensure reproducibility. The MLP weights are initialized using Kaiming initialization for hidden layers and zero-initialization for the output layer, to start from an identity mapping, with zero initial deformation of control points.

The numerical validation is conducted on a modified version of KVLCC2 [29] and all results are reported in model scale. The primary objective of this numerical experiment was to assess the solver's capability to handle competing geometric constraints. The optimization problem was formulated to reduce the Wetted Surface Area (WSA) and the maximum Draft (minimum z coordinate) while strictly conserving the volume and the longitudinal position of the Center of Buoyancy (COB). The specific targets were defined relative to the initial geometric properties:

- Volume Conservation: $V_{target} = 1.0 \cdot V_0$

[2]The input geometry was parsed using the *pyiges* library [29].

[3]Hardware specifications: Intel Core i7 9th Gen H-series, 16 GB DDR4 RAM, NVIDIA GeForce GTX 1650 Mobile; OS: Ubuntu 24.04 LTS.

Algorithm 1: Optimization Loop

Input: Reference grid C^0, Quadrature rule (Ξ, W).
Output: Optimized parameters $\boldsymbol{\theta}^*$.
Initialize MLP weights $\boldsymbol{\theta}_0$
Compute reference configuration $\mathbf{x}^0(\Xi)$

while $t < T_{\max}$ **do**

`// Neural Deformation Field`

$\Delta\tilde{C} \leftarrow \Phi_{\boldsymbol{\theta}_t}(C^0)$

$\Delta C \leftarrow \mathcal{P}(\Delta\tilde{C})$

$C \leftarrow C^0 + \Delta C$

`// Geometric Analysis`

for $q = 1, \ldots, N_q$ **do**

$\mathbf{x}_q \leftarrow \mathbf{S}(\boldsymbol{\xi}_q; C)$

$\mathbf{t}_{u,q} \leftarrow \partial_u \mathbf{S}(\boldsymbol{\xi}_q; C)$

$\mathbf{t}_{v,q} \leftarrow \partial_v \mathbf{S}(\boldsymbol{\xi}_q; C)$

$\mathbf{m}_q \leftarrow \mathbf{t}_{u,q} \times \mathbf{t}_{v,q}$

end

`// Functional Assembly`

$\mathcal{R}_V \leftarrow \left(\frac{\sum_q (z_q - z_{ref})(\mathbf{m}_q \cdot \mathbf{e}_z) w_q - V_{target}}{V_{target}} \right)^2$

$\mathcal{R}_A \leftarrow \left(\frac{\sum_{z_q \le z_{ref}} \|\mathbf{m}_q\| w_q - A_{target}}{A_{target}} \right)^2$

$\mathcal{R}_C \leftarrow \left(\frac{\sum_q x_q (z_q - z_{ref})(\mathbf{m}_q \cdot \mathbf{e}_z) w_q}{x_{COB,target} \sum_q (z_q - z_{ref})(\mathbf{m}_q \cdot \mathbf{e}_z) w_q} - 1 \right)^2$

$\mathcal{R}_D \leftarrow \sum_q \left(\max(0, Z_{min} - z_q) \right)^2$

$\mathcal{E}_{reg} \leftarrow \sum_q \left(e^{\gamma \|\mathbf{x}_q - \mathbf{x}_q^0\|^2} - 1 \right) \|\mathbf{m}_q\| w_q$

`// Parameter Update`

$\mathcal{L}(\boldsymbol{\theta}_t) \leftarrow \lambda_V \mathcal{R}_V + \lambda_A \mathcal{R}_A + \lambda_C \mathcal{R}_C + \lambda_D \mathcal{R}_D + \lambda_{reg} \mathcal{E}_{reg}$

$\boldsymbol{\theta}_{t+1} \leftarrow \text{Optimizer}(\boldsymbol{\theta}_t, \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t))$

end

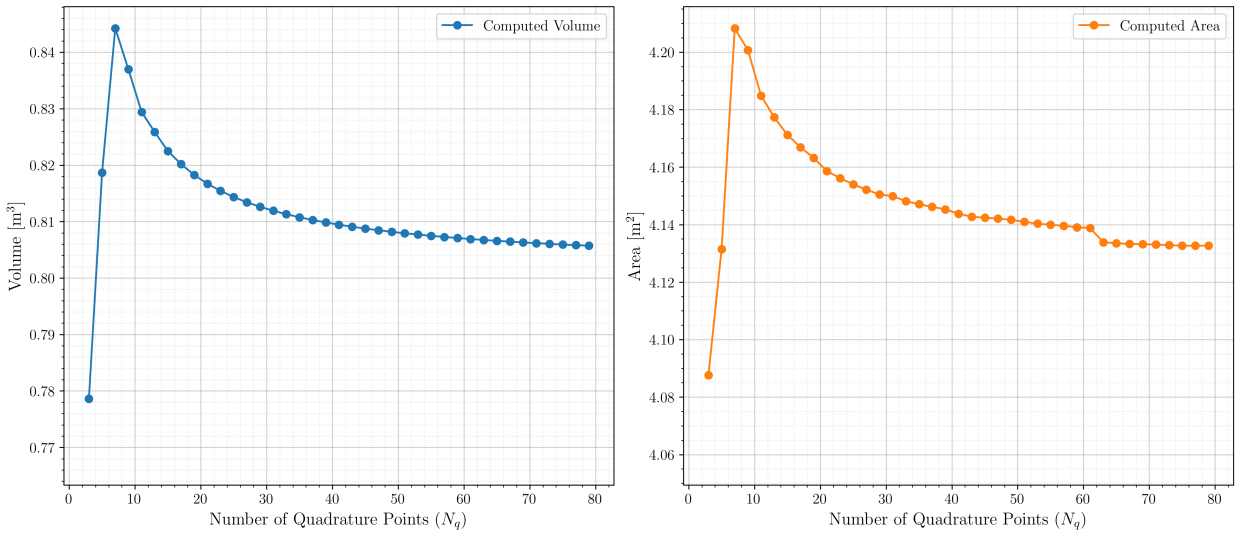


Figure 3: Convergence and stability analysis of geometric integrals via Gauss-Legendre quadrature.
Left panel: Volumetric stability. **Right panel:** Convergence of the Wetted Surface Area (WSA). Both metrics motivate the selection of $N_q = 18$ per dimension as a sufficient trade-off resolution, ensuring that numerical integration noise remains negligible compared to the analytical gradients driving the neural deformation field.

- Wetted Area Minimization: $A_{target} = 0.97 \cdot A_0$ (3% reduction)
- Draft Constraint: $Z_{min}^{target} = 0.95 \cdot Z_{min}^0$ (5% reduction in depth)
- Centroid Preservation: $x_{COB}^{target} = x_{COB}^0$

This setup represents a preliminary design scenario where a vessel's efficiency is improved (lower friction drag via area reduction) and operational flexibility is increased (lower draft), without sacrificing capacity (volume).

4.2.1. Numerical Convergence

The optimization was performed over 2000 epochs and Table 2 details the optimization parameters utilized. The processed CAD model represents a symmetric half-hull. To preserve the tensor-product structure required for exact numerical quadrature, the geometry utilizes the native untrimmed NURBS patches, intentionally omitting the minor deck trimming present in the official KVLCC2 benchmark, having a slightly different area and volume compared to the original hull.

As presented in Table 3, the barrier constraint was satisfied within an absolute algorithmic tolerance ($1e-3$). The final minimum z value reached -0.34085, deemed acceptable. The solver achieved an area of approximately 4.011, reaching the target of 4.005 (approximately 3% reduction). The final relative error against the target was +0.159%, corresponding to a net physical reduction of 2.845% from the initial state. Volume conservation was achieved with high accuracy, resulting in a final deviation of -0.076%. This indicates that the solver successfully balanced the volume loss induced by the draft reduction through deformation in other regions.

Regarding the Center of Buoyancy (COB), the longitudinal position (x) was preserved with negligible error (0.00001 m). However, a total 3D shift of 0.01569 m was observed, primarily driven by vertical (z) and transverse (y) displacements. This shift is consistent with the geometric transformation required to maintain constant volume while reducing draft: as the bottom is raised, volume is redistributed laterally and vertically, naturally elevating the vertical center of buoyancy. The observed total shift in the total center of buoyancy does not compromise the hydrostatic equilibrium of the hull. The optimization of a symmetric semi-geometry strictly enforces transverse symmetry ($y = 0$), preventing any induced heel. Furthermore, the conservation of the immersed volume ensures that the design waterline remains consistent, independent of the variation in the vertical center of buoyancy (z_{COB}).

Table 2: Optimization parameters and configuration settings for the *KVLCC2 modified* optimization.

Parameter	**Value**
Input File	`kvlcc2m.iges`
Seed	911
Reference Plane (Z_{ref}) [m]	0.0
Quadrature Points (N_q)	18
Optimization Epochs	2000
MLP Architecture	
Hidden Neurons	64
Hidden Layers	5
Learning Rate	$1 \cdot 10^{-3}$
Loss Weights	
λ_{vol} (Volume)	2.0
λ_{area} (Area)	1.0
λ_{depth} (Draft Barrier)	1.0
λ_{cen} (Centroid)	1.0
λ_{reg} (Regularization)	$1 \cdot 10^{-4}$
γ_{reg} (Exp. Decay)	200.0

Table 3: Optimization results for the modified KVLCC2 (seed 911). Volume and Area errors are relative [%]. Centroid deviations are in meters.

Metric	**Initial**	**Target**	**Final**	**Error**
Volume [m^3]	0.80165	0.80165	0.80104	-0.076 %
Wetted Area [m^2]	4.12894	4.00507	4.01145	+0.159 %
Max Draft ($z_{\min}$) [m]	−0.35873	−0.34079	−0.34085	-0.018 %
Absolute Geometric Deviations				
Centroid x [m]	−0.22528	−0.22528	−0.22527	0.00001 [m]
Centroid y [m]	0.22813	0.22813	0.23973	0.01160 [m]
Centroid z [m]	−0.17117	−0.17117	−0.16060	0.01057 [m]
Centroid Shift (L_2) [m]	–	–	–	0.01569 [m]

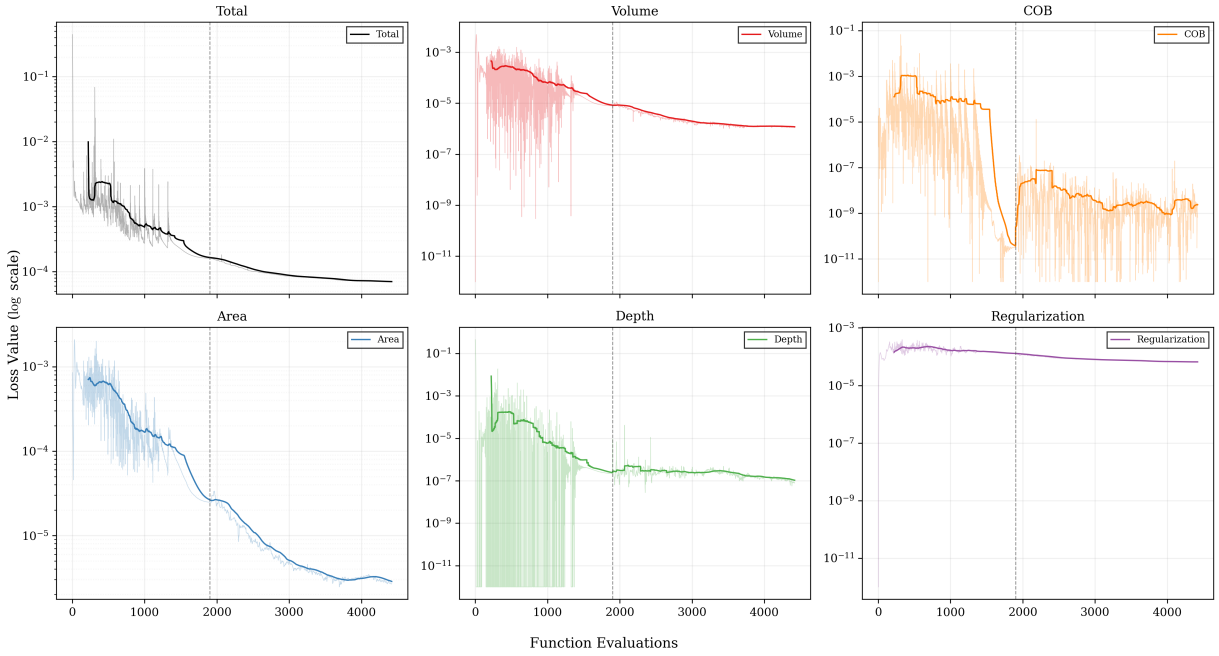


Figure 4: Optimization convergence analysis (Seed= 911 and $\lambda_{reg} = 1e^{-4}$). The plots illustrate the evolution of the global loss and specific geometric constraint losses against the number of function evaluations. Raw oscillation data is shown with high transparency, overlaid by a solid line representing a moving average to enhance trend visibility. The vertical dashed line indicates the switch from the stochastic exploration phase (AdamW) to the deterministic second-order refinement phase (L-BFGS). While the optimization was executed over 2000 epochs, the accumulation of more than 4000 function evaluations is a direct consequence of the L-BFGS algorithm, which inherently requires multiple evaluations per iteration for line search routines.

4.2.2. Geometric Deformation Analysis

Figure 5 shows the render comparison between the original surface and the deformed one. The solver satisfied the draft constraint by effectively modifying the bottom nodes. To counteract the volume loss caused by the reduced draft, the neural network generates a transverse expansion the hull beam slightly in the midship section and reshape the bulbous bow, as suggested by the increase in the y-component of the centroid. This non-rigid deformation demonstrates the capability of the MLP-based parametrization to discover complex shape compensations.

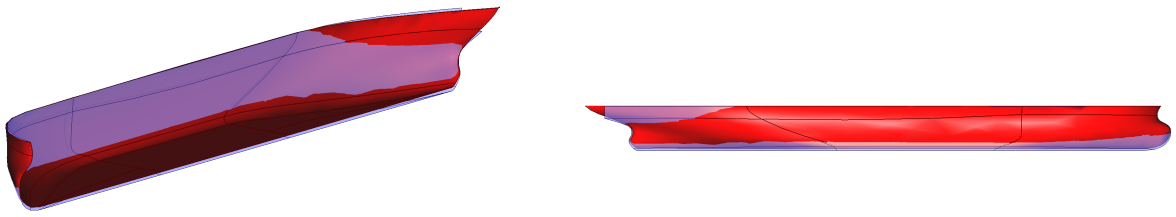


Figure 5: Visualization of the optimized geometric deformation. Overlay of the initial hull (blue) and the optimized hull (red). Note the significant beam expansion at the midship section and the reshaping of the bulbous bow, generated by the network to compensate for the volume loss induced by the strict draft reduction constraint (Z_{min}). (Seed = 911 and $\lambda_{reg} = 1e^{-4}$).

4.3. Benchmark against Free-Form Deformation (FFD)

Free-Form Deformation (FFD) represents a standard methodology for CAD shape optimization and is utilized as the baseline to evaluate the proposed neural parametrization. The Spline architecture is still used, but the MLP deformation net is replaced by a FFD. This is implemented as a differentiable module computing the displacement field

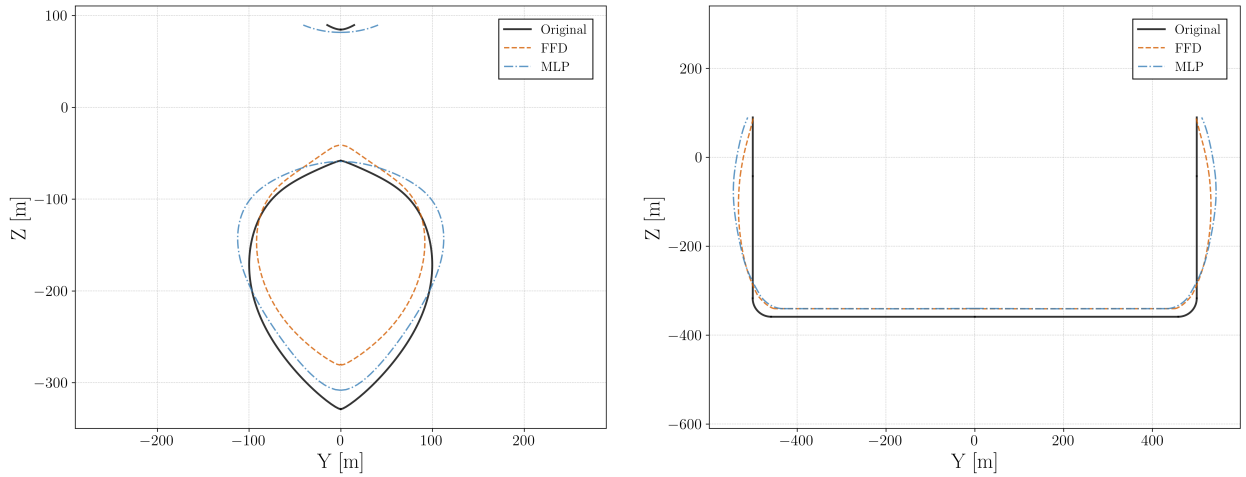
via Bernstein polynomials over a global spatial bounding box that encapsulates the NURBS control points [8]. An anisotropic grid resolution of $24 \times 8 \times 8$ is utilized, introducing 4608 independent degrees of freedom. To ensure a rigorous comparative analysis, the objective function, geometric barrier, integral constraints, regularization penalties, and the dual stage optimization scheme remain identical to the MLP configuration.

Optimization metrics for the FFD baseline are detailed in Table 4. The FFD solver achieves numerical convergence, satisfying the minimum draft barrier with a relative error of -0.012% and conserving the displaced volume to within -0.036%.

Table 4: Optimization results for the modified KVLCC2 using Free Form Deformation (seed 911). Volume and Area errors are relative [%].

Metric	**Initial**	**Target**	**Final**	**Error**
Volume [m^3]	0.80165	0.80165	0.80136	-0.036 %
Wetted Area [m^2]	4.12894	4.00507	4.00909	+0.100 %
Max Draft ($z_{\min}$) [m]	−0.35873	−0.34079	−0.34083	-0.012 %
Absolute Geometric Deviations				
Centroid x [m]	−0.22528	−0.22528	−0.22527	0.00000 [m]
Centroid y [m]	0.22813	0.22813	0.23848	+0.01035 [m]
Centroid z [m]	−0.17117	−0.17117	−0.16190	+0.00926 [m]
Centroid Shift (L_2) [m]	–	–	–	0.01389 [m]

Furthermore, the FFD methodology imposes an intrinsic resolution trade-off between geometric fidelity and computational cost: coarse FFD grids lack the local support necessary to capture and deform high-frequency geometric features, such as the bulbous bow, whereas dense grids exponentially increase the degrees of freedom. This unregularized behavior is incompatible with the objectives of this work, as the resulting geometric degradation compromises the topological validity and surface smoothness required by industrial standards. Unlike the non-linear MLP mapping, which adapts simultaneously to global structural modifications and local morphing requirements without grid dependencies, the extrinsic FFD grid is agnostic to the underlying topology. Consequently, it may more easily induce optimization stagnation when the required geometric transformations do not align with the principal axes of the control lattice.



(a) Bow Section (1/20)

(b) Midship Section (11/20)

Figure 6: Comparative cross-sectional analysis of the modified KVLCC2 hull geometry optimized via Free-Form Deformation (FFD) and the proposed neural parametrization (MLP). Same optimisation parameters.

As observed in Figure 6, both methodologies produce comparable global deformation trends, in particular in the

midship region. However, the MLP architecture exhibits superior flexibility in resolving complex shape compensations. This capability is explicitly evident in the bow section (Figure 6a), where the neural parametrization yields a more distinctive geometric profile compared to the conventional deformation induced by the FFD approach.

4.4. Benchmark against Direct Control Points Optimization

To quantify the regularization effect of the MLP Neural parametrization, a baseline experiment was conducted where the control points $\mathbf{C}_{i,j}$ were optimized directly as independent variables, bypassing the neural parameterization ($\Phi_{\boldsymbol{\theta}}$) from the pipeline. The optimization parameters are the same as above. While this approach allows for a theoretical zero-residual solution due to the maximized degrees of freedom, the resulting geometry exhibited severe topological degradation.

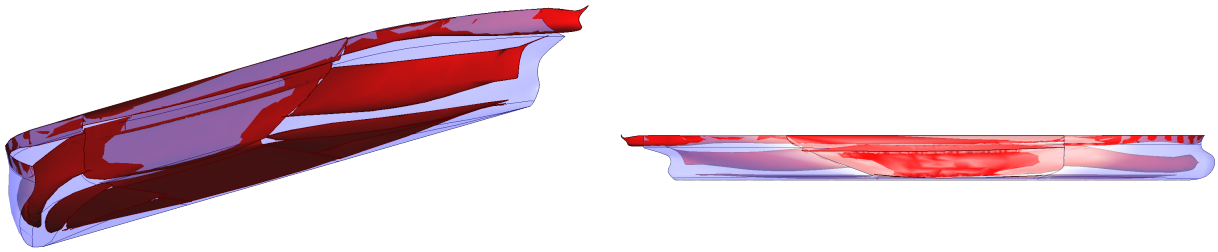
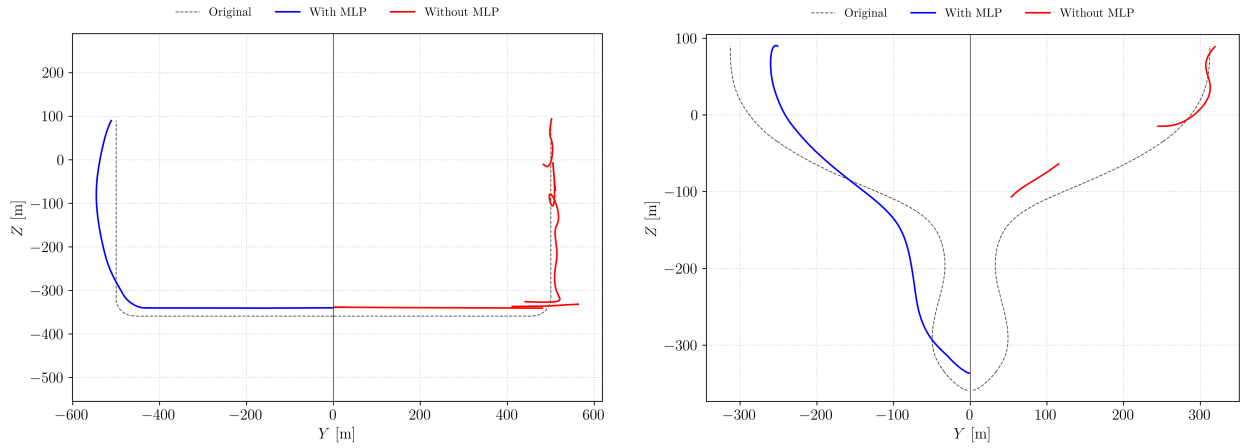


Figure 7: Impact of the parametrization strategy on geometric quality. The direct optimization of control points lacks regularization, resulting in high-frequency artifacts and loss of C^0 continuity at patch interfaces (visible as surface gaps). The blue geometry is the original undeformed surface.



(a) Midship Station (10/20)

(b) Aft Section (18/20)

Figure 8: Comparative cross-sectional analysis at Station 10 and Station 18 over 20 total sections. The negative Y domain ($Y < 0$) visualizes the proposed MLP-driven deformation (blue), demonstrating strict adherence to topological constraints and the preservation of global hull continuity. In contrast, the positive Y domain ($Y > 0$) depicts the unregularized direct control point optimization (red), characterized by the breakdown of C^0 continuity and the macroscopic detachment of adjacent NURBS patches relative to the reference hull geometry (dashed grey).

As illustrated in Figure 7 and Figure 8, the direct optimization lacks the regularisation effect of the neural network. Consequently, the control points drifted incoherently, resulting in high-frequency geometric noise and, critically, the violation of C^0 continuity at the patch interfaces. In the absence of the global deformation field provided by $\mathcal{N}_\theta$, maintaining watertight integrity would require the explicit imposition of computationally expensive equality constraints for every shared boundary node.

While direct optimization treats each control point as an independent degree of freedom, making it prone to high-frequency artifacts, the MLP parametrization inherently favors low-frequency, smooth solutions due to the spectral

bias of neural networks [30]. This effectively correlates the movement of spatially adjacent control points without requiring explicit smoothing penalties, consistent with findings in structural optimization [25].

Furthermore, the neural parametrization offers an advantage in terms of dimensionality. In a direct optimization setting, the number of degrees of freedom scales linearly with the number of control points, which can reach thousands for high-fidelity industrial models ($N_{dof} \approx 3 \times N_{cp}$). Conversely, the MLP parameter space is fixed by the network architecture ($N_{weights}$), independent of the geometric resolution, allowing for better control over memory consumption.

4.5. Robustness Analysis and Variance

To assess the dependence of the optimization outcome on the stochastic initialization of the neural network weights, the framework was executed using 20 distinct random seeds, using the same other optimisation parameters of Table 2. Table 5 summarizes the final geometric properties for each run.

Table 5: Hull optimization results. Statistical performance aggregated over 20 random seeds compared to *Initial* and *Target* values. Reported values represent Mean ± Standard Deviation.

Metric	**Initial**	**Target**	**Statistical Analysis** (20 Seeds)
Final Volume [m^3]	0.80165	0.80165	0.80086 ± 0.00064
Volume Error [%]	–	0.0	−0.099 ± 0.079
Wetted Area [m^2]	4.12894	4.00507	4.01630 ± 0.01268
Area Error [%]	–	0.0	0.280 ± 0.317
Physical Area Delta [%]	–	−3.0	−2.728 ± 0.307
Max Draft ($z_{\min}$) [m]	−0.35873	−0.34079	−0.34090 ± 0.00003
Draft Dev. (from Limit) [%]	–	0.0	−0.032 ± 0.010
Absolute Geometric Deviations			
Centroid Shift (L_2) [m]	–	0.0	0.01566 ± 0.00112
Centroid Shift (x) [m]	–	0.0	0.00001 ± 0.00001

Metric Definitions. Volume and wetted area errors are computed as relative deviations from targets. The draft deviation is evaluated relative to the lower bound ($z_{\min} \geq z_{\text{limit}}$); negative values indicate a slight violation of the strict bound, tolerated by a small numerical relaxation of the depth barrier formulation.

Despite the different starting configurations of the MLP ($\mathcal{N}_{\theta_0}$), the solver consistently converges, exhibiting interesting marginal deviations (Figure 9). The standard deviation of the final volume error is very low ($\sigma_{vol} \approx 0.08\%$), and the wetted surface area reduction remains consistent across all runs (approx. $2.73\% \pm 0.31\%$). Notably, the draft constraint exhibits the highest stability, with the final minimum vertical coordinate z_{min} varying by less than $4 \cdot 10^{-5}$ m across trials. This confirms that the penalty-based barrier function $\mathcal{L}_{depth}$ effectively enforces the hard geometric limit regardless of the initial exploration path. This is due also to the highest value of λ in the composite loss function (Table 2). The longitudinal centroid shift shows minor fluctuations (0.0 m to $2 \cdot 10^{-5}$ m), suggesting that while the macroscopic hydrostatic properties are statistically equivalent, the network identifies slightly different local deformation maps to satisfy the global integral constraints, described in Table 5.

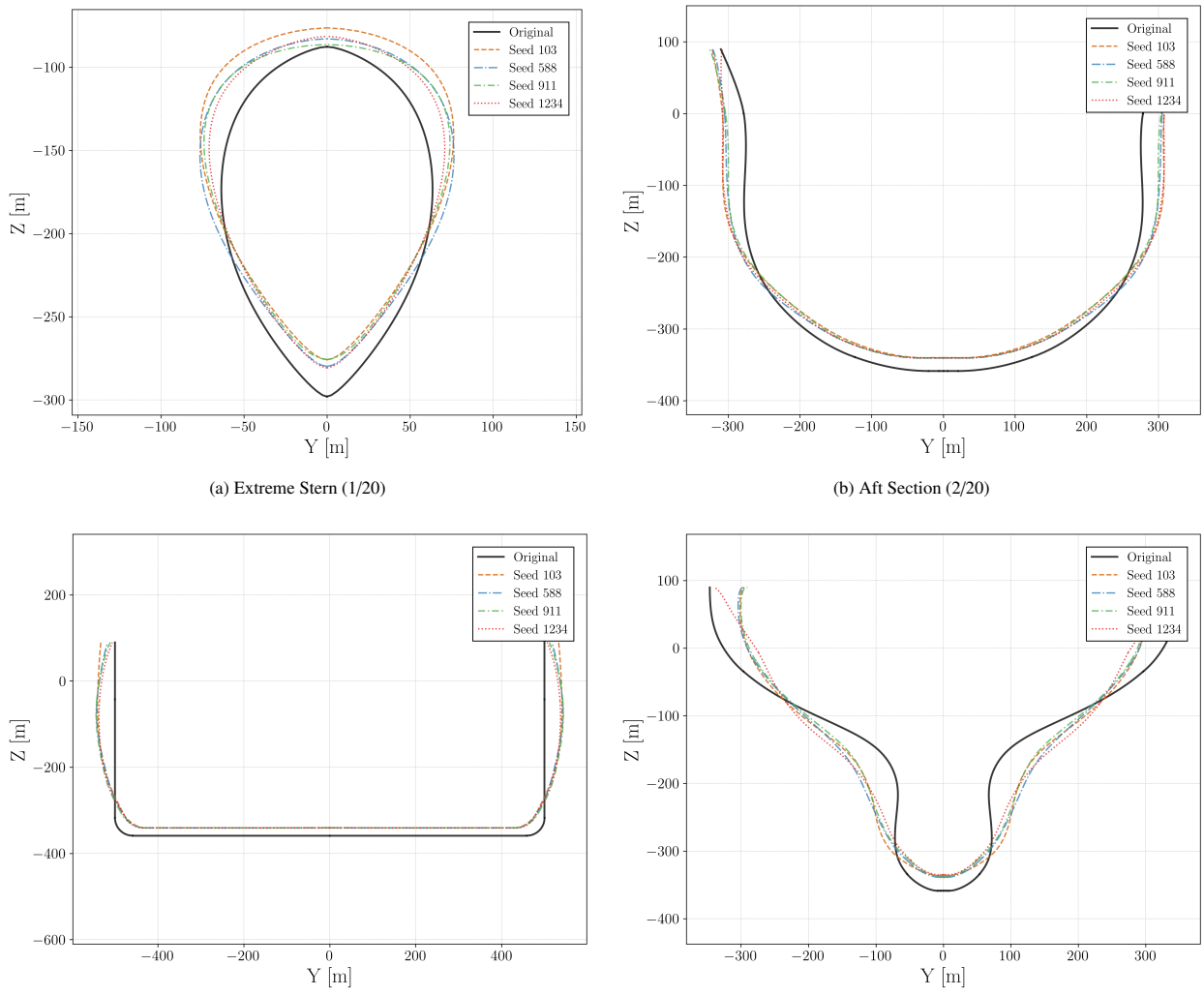
The $N = 20$ geometric sections Γ_k found in Figure 9 are derived by computing the intersection between the NURBS B-Rep surface $\partial\Omega$ and a set of equidistant transverse planes Π_k:

$$\Gamma_k = \partial\Omega \cap \Pi_k, \quad \Pi_k = \{\mathbf{x} \in \mathbb{R}^3 \mid x = x_k\} \tag{17}$$

where x_k is linearly spaced along the longitudinal axis of the hull.

4.6. Sensitivity Analysis of Regularization

The sensitivity of the optimization framework to the regularization weight λ_{reg} was investigated by varying the parameter across four orders of magnitude, from 10^{-5} to 10^{-2}, averaging the results among 20 different seeds. Table 6



(a) Extreme Stern (1/20)

(b) Aft Section (2/20)

(c) Midship Section (11/20)

(d) Bow Section (18/20)

Figure 9: Robustness analysis of the neural parametrization with respect to stochastic initialization. Comparison of cross-sections at various longitudinal stations for multiple random seeds.

Table 6: Hull Optimization: Regularization sensitivity analysis (Mean ± SD, $n = 20$ for every λ value)

Metric	**Initial**	**Target**	$\lambda = 10^{-5}$	$\lambda = 10^{-4}$	$\lambda = 10^{-3}$	$\lambda = 10^{-2}$
Volume [m^3]	0.80165	0.80165	0.80153 ± 0.00002	0.80086 ± 0.00064	0.79850 ± 0.00088	0.79010 ± 0.00028
Vol. Error [%]	–	0.0	−0.015 ± 0.003	−0.099 ± 0.079	−0.392 ± 0.110	−1.440 ± 0.035
Area [m^2]	4.12894	4.00507	4.00588 ± 0.00275	4.01630 ± 0.01268	4.03457 ± 0.01025	4.07731 ± 0.00347
Area Err. [%]	–	0.0	+0.020 ± 0.069	+0.280 ± 0.317	+0.737 ± 0.256	+1.804 ± 0.087
Δ Area [%]	–	-3.0	−2.980 ± 0.067	−2.728 ± 0.307	−2.285 ± 0.248	−1.250 ± 0.084
Max Draft [m]	-0.35873	-0.34079	−0.34085 ± 0.00002	−0.34090 ± 0.00003	−0.34102 ± 0.00005	−0.34131 ± 0.00005
Draft Dev. [%]	–	0.0	−0.018 ± 0.004	−0.032 ± 0.010	−0.068 ± 0.014	−0.151 ± 0.014
Shift L_2 [m]	–	0.0	0.01761 ± 0.00078	0.01566 ± 0.00112	0.01334 ± 0.00061	0.00980 ± 0.00006
Shift x [m]	–	0.0	0.00000 ± 0.00000	0.00001 ± 0.00001	0.00003 ± 0.00001	0.00014 ± 0.00001

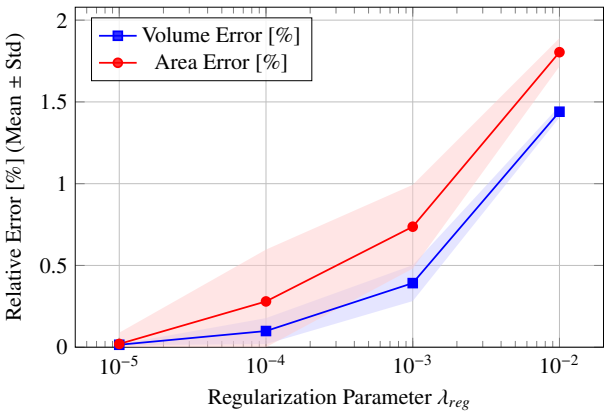


Figure 10: Sensitivity analysis of the geometric optimization performance with respect to the regularization weight λ_{reg} for the KVLCC2 hull. Solid lines represent the mean relative error [%] for volume (blue squares) and wetted surface area (red circles) aggregated over 20 stochastic initializations. Shaded regions denote the ±1 standard deviation interval, illustrating the convergence of the neural parametrization.

and Figure 10 present the resulting metrics, revealing a fundamental trade-off between the satisfaction of hydrodynamic constraints and the preservation of the initial hull geometry.

Increasing λ_{reg} progressively stiffens the deformation field, reducing the mean centroid shift (L_2 norm) from 0.018 m at $\lambda_{reg} = 10^{-5}$ to 0.010 m at $\lambda_{reg} = 10^{-2}$. However, this geometric rigidity severely compromises the solver's ability to reach the target objectives. At the highest regularization level ($\lambda_{reg} = 10^{-2}$), the mean displaced volume error degrades to -1.44%, a value incompatible with the precision required for hydrodynamic design. In contrast, regimes with $\lambda_{reg} \leq 10^{-4}$ maintain volume and wetted area relative errors below 0.3%.

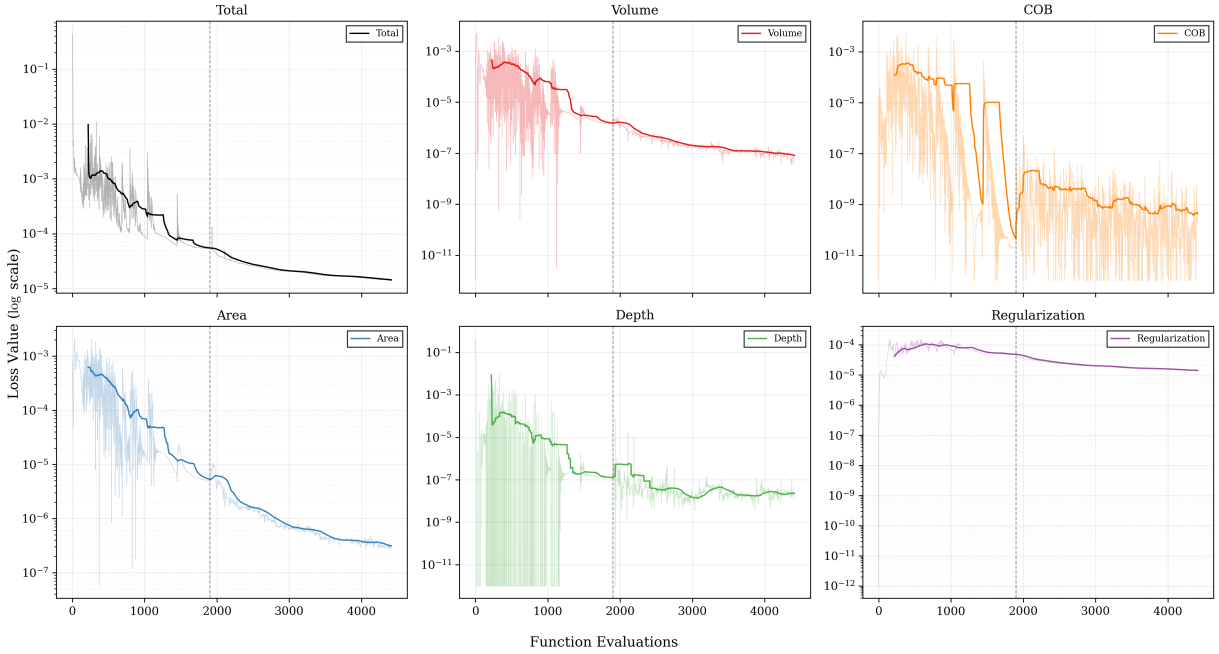
The underlying optimization dynamics are visualized in Figure 11, which contrasts the optimisation histories of the limiting regimes. In a low regularization case ($\lambda_{reg} = 10^{-5}$), the transition from the AdamW [31] optimizer to the L-BFGS [32] algorithm triggers a reduction in total residuals of approximately two orders of magnitude, indicating successful convergence to the physical constraints. Conversely, the stiff regime ($\lambda_{reg} = 10^{-2}$) is dominated by the regularization gradient throughout the process. Based on these findings, $\lambda_{reg} = 10^{-4}$ is identified a possible optimal compromise to ensure negligible physical errors while preventing mesh degeneracy.

The numerical trade-off observed is physically shown by the cross-sectional analysis in Figure 12. At the midship section (Figure 12c), satisfying the draft constraint (z_{min}) forces a vertical displacement of the keel. To conserve the total volume, the solver must redistribute the displaced mass by expanding the hull beam. As observed in the cross-sectional analysis (Figure 12c), the unconstrained volume compensation alters the verticality of the ship sides. While hydrostatically viable for an oil carrier, this geometric deformation could introduce inefficiencies in vessel classes requiring vertical side walls, such as container ships. Further dimensional constraints, analogous to the enforced draft barrier, can be directly integrated into the optimization objective to preserve specific structural features based on the designer's operational requirements.

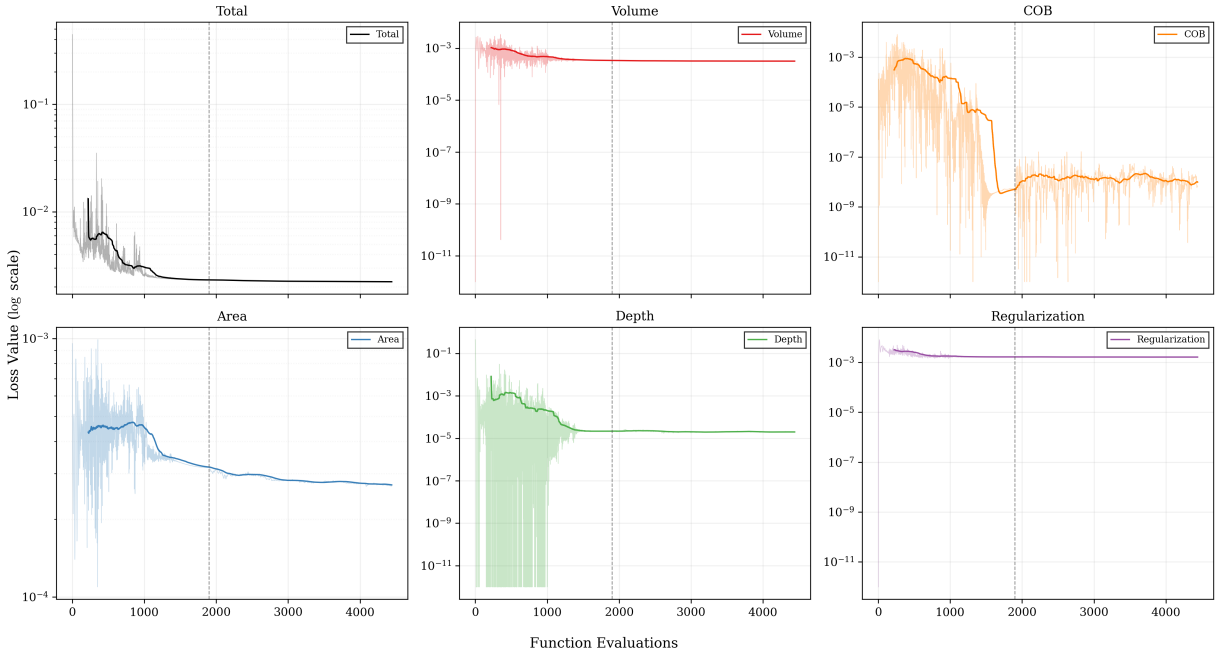
The high regularization regime ($\lambda_{reg} = 10^{-2}$) imposes excessive stiffness on the control points, preventing this lateral expansion. This results in the significant volume loss discussed above. Conversely, the low regularization regime ($\lambda_{reg} = 10^{-5}$) grants the neural network sufficient flexibility to identify a non-rigid compensation mechanism, widening the hull sections while maintaining surface fairness. Furthermore, Figures 12a, 12b and 12d demonstrate that this deformation is not localized but globally distributed towards the stern and bow, ensuring the preservation of the longitudinal center of buoyancy (x_{COB}).

5. Conclusions

This work established an end-to-end differentiable framework for the shape optimization of parametric surfaces, implemented within the PINA ecosystem. By leveraging analytical B-Spline derivatives for the computation of the Jacobian determinant, the proposed method enables fast steps for integral constraints problems. This optimisation

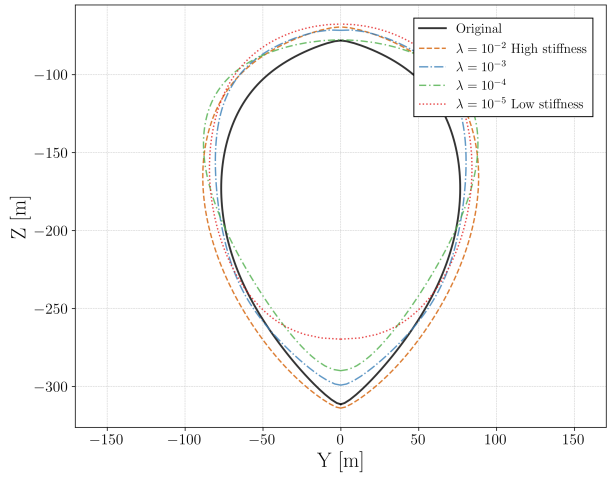


(a) Example of Low Regularization ($\lambda = 10^{-5}$). Physics driven convergence.

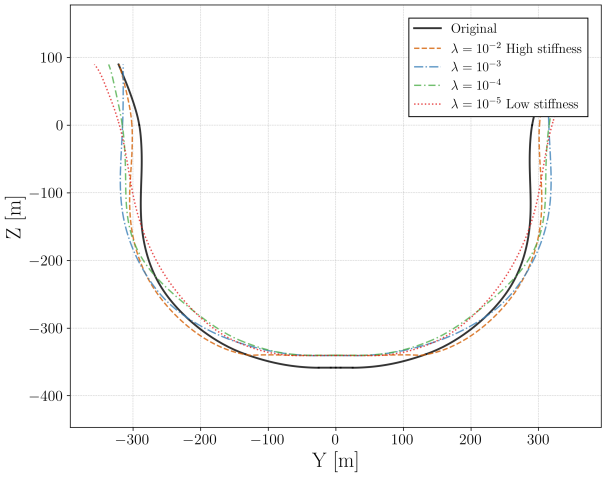


(b) Example of High Regularization ($\lambda = 10^{-2}$). Regularization dominated regime.

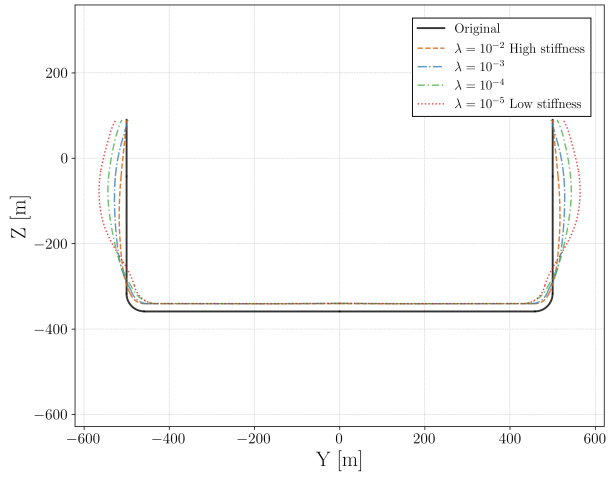
Figure 11: Comparison of optimisation dynamics for two representative regularization regimes (Seed 911). The vertical dashed line marks the switch from Adam to L-BFGS optimizer. (a) Low λ_{reg} allows the second-order optimizer to effectively minimize physical residuals. (b) High λ_{reg} leads to stagnation as the optimization landscape is dominated by the regularization penalty.



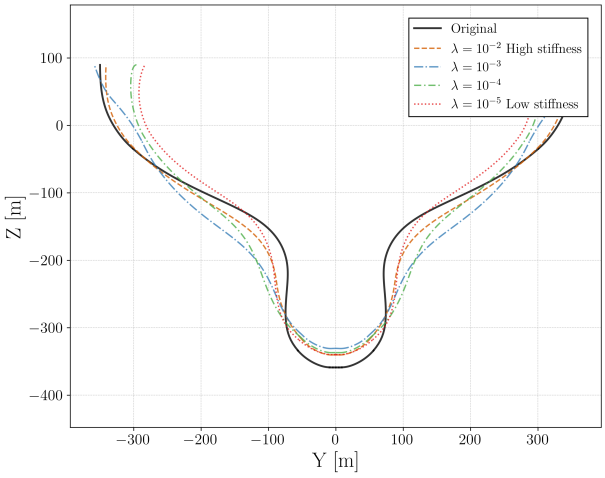
(a) Extreme Stern (1/20)



(b) Aft Section (2/20)



(c) Midship Section (11/20)



(d) Bow Section (18/20)

Figure 12: *Sensitivity of geometric deformation to regularization stiffness.* Cross-sections at four representative stations. **(c)** At midship, the *Low Stiffness* regime (orange, $\lambda_{reg} = 10^{-5}$) enables the necessary beam expansion to compensate for the draft reduction (Z_{min} constraint), ensuring volume conservation within numerical tolerance. In contrast, the *High Stiffness* regime (red, $\lambda_{reg} = 10^{-2}$) locks the geometry to the original manifold, leading to the volume violations. The deformation field is global: mass redistribution is evident along the entire hull to maintain the longitudinal center of buoyancy (x_{COB}). The random seed used is 911.

formulation avoids the geometrical discretization overhead and gradient inconsistencies inherent to traditional mesh-based analysis workflows, and more flexibility than traditional Free Form Deformations standard approaches.

A key contribution is the introduction of a neural deformation strategy, where a Multi-Layer Perceptron (MLP) parametrizes the global displacement field of the control net. Comparative benchmarks against direct control point optimization demonstrate that the MLP architecture functions as an implicit regularizer. This regularization is critical for preserving C^0 continuity across multi-patch interfaces, effectively preventing the geometric degradation observed in direct approaches.

Numerical validation on the hull geometry confirmed the framework's efficacy in handling competing geometric constraints. The solver successfully enforced a 5% draft reduction constraint and achieved an approximate 2.8% average reduction in wetted surface area (against a 3% target), while maintaining the longitudinal geometry centroid. The method achieved volume conservation with a mean relative error below 0.1%, producing geometries with interface continuity. Sensitivity analysis identified a regularization regime ($\lambda_{reg} \approx 10^{-4}$) ensuring convergence to valid design candidates.

The intrinsic generality of the proposed framework allows for its extension to heterogeneous engineering domains, provided that the integral geometric constraints and boundary conditions are appropriately adapted to the specific physical problem.

Future research can focus on two primary directions. First, the framework can be extended to support trimmed NURBS surfaces to address arbitrary industrial topologies defined by non-tensor product boundaries. Second, the pipeline can incorporate PINN solvers to resolve Partial Differential Equations (PDEs) directly on the evolving manifold. Leveraging the PINA infrastructure, this extension aims to transition from geometric constraint satisfaction to functional optimization, targeting physical performance metrics such as hydrodynamic drag or structural stress. In addition, the actual discrete integration can be replaced by the analytical formulation of B-Spline polynomial integration.

Acknowledgements

F.T., G.A.D., N.D. and G.R. acknowledge the support provided by the European Union – NextGenerationEU, in the framework of the iNEST – Interconnected Nord-Est Innovation Ecosystem (iNEST ECS00000043 – CUP G93C22000610007) project and its CC5 Young Researchers initiative. G.C. acknowledges the support provided by the European Union – NextGenerationEU, within the framework of the National Recovery and Resilience Plan (PNRR), Mission 4, Component 2, Investment 3.3, through a PhD scholarship at Scuola Internazionale Superiore di Studi Avanzati di Trieste (CUP G93C24000530001), developed in collaboration with Fincantieri S.p.A. G.R. acknowledges the support provided by PRIN "FaReX - Full and Reduced order modelling of coupled systems: focus on non-matching methods and automatic learning" project. The views and opinions expressed are solely those of the authors and do not necessarily reflect those of the European Union, nor can the European Union be held responsible for them. G.A.D. and G.R. would like to acknowledge INdAM–GNCS.

References

[1] V. Srinivasan, An integration framework for product lifecycle management, Computer-aided design 43 (5) (2011) 464–478.

[2] R. Sudarsan, S. J. Fenves, R. D. Sriram, F. Wang, A product information modeling framework for product lifecycle management, Computer-aided design 37 (13) (2005) 1399–1411.

[3] Z. Bi, X. Wang, Computer aided design and manufacturing, John Wiley & Sons, 2020.

[4] K. J. Versprille, Computer-aided design applications of the rational b-spline approximation form., Syracuse University, 1975.

[5] A. Ivagnes, N. Demo, G. Rozza, A shape optimization pipeline for marine propellers by means of reduced order modeling techniques, International Journal for Numerical Methods in Engineering (2024). arXiv:2305.07515, doi:10.1002/nme.7426.

[6] N. Demo, M. Tezzele, A. Mola, G. Rozza, Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing, Journal of Marine Science and Engineering 9 (2) (2021). arXiv:2101.03781, doi:10.3390/jmse9020185.
URL https://www.mdpi.com/2077-1312/9/2/185

[7] D. Villa, F. Furcas, J. O. Pralits, G. Vernengo, S. Gaggero, An effective mesh deformation approach for hull shape design by optimization, Journal of Marine Science and Engineering 9 (10) (2021) 1107.

[8] T. W. Sederberg, S. R. Parry, Free-form deformation of solid geometric models, in: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, 1986, pp. 151–160.

[9] D. Wei, L. Wang, J. Yang, X. Sun, X. Ji, X. Liu, Hull form optimization with a new three-dimensional deformation strategy, Scientific Reports 15 (1) (2025) 41804.

[10] S. Li, F. Zhu, X. Hou, Q. Ni, Application of mesh deformation and adaptive method in hullform design optimization, Journal of Marine Science and Technology 27 (1) (2022) 566–575.

[11] S. Wang, Y. Kim, Cfd-based hull optimization in calm water using adaptive grid deformation method, Journal of Hydrodynamics 36 (2) (2024) 241–262.

[12] A. Mola, N. Demo, M. Tezzele, G. Rozza, Chapter 17: Geometrical Parametrization and Morphing Techniques with Applications, 2022, pp. 345–364. doi:10.1137/1.9781611977257.ch17.
URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977257.ch17

[13] J. Lee, S. Hube, S. Elgeti, Neural networks vs. splines: advances in numerical extruder design, Engineering with Computers 40 (2) (2024) 989–1004.

[14] S. Khan, K. Goucher-Lambert, K. Kostas, P. Kaklis, Shiphullgan: A generic parametric modeller for ship hull design using deep convolutional generative model, Computer Methods in Applied Mechanics and Engineering 411 (2023) 116051.

[15] N. J. Bagazinski, F. Ahmed, Shipgen: A diffusion model for parametric ship hull generation with multiple objectives and constraints, Journal of Marine Science and Engineering 11 (12) (2023) 2215.

[16] G. Padula, F. Romor, G. Stabile, G. Rozza, Generative models for the deformation of industrial shapes with linear geometric constraints: Model order and parameter space reductions, Computer Methods in Applied Mechanics and Engineering 423 (2024) 116823.

[17] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics 378 (2019) 686–707.

[18] Y. Sun, U. Sengupta, M. Juniper, Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry, Computer Methods in Applied Mechanics and Engineering 411 (2023) 116042.

[19] S. Jalui, J. O'Connor, Y. Xuan, G. Manogharan, Developing novel restrictive design for additive manufacturing (DfAM) constraints for NURBS-based adjoint shape optimization for metal AM, Graphical Models 142 (2025) 101304.

[20] F. Mohebbi, B. Evans, On an exact step length in gradient-based aerodynamic shape optimization, Fluids 5 (2) (2020) 70.

[21] A. Moola, A. Balu, A. Krishnamurthy, A. Pawar, THB-Diff: a GPU-accelerated differentiable programming framework for THB-splines, Engineering with Computers 40 (6) (2024) 3477–3493.

[22] S. Tillmann, D. Hilger, N. Hosters, S. Elgeti, Shape-optimization of extrusion-dies via parameterized physics-informed neural networks, PAMM 23 (4) (2023) e202300203.

[23] Z. Zhang, C. Lin, B. Wang, Physics-informed shape optimization using coordinate projection, Scientific Reports 14 (1) (2024) 6537.

[24] D. Coscia, A. Ivagnes, N. Demo, G. Rozza, Physics-informed neural networks for advanced modeling, Journal of Open Source Software 8 (87) (2023) 5352.

[25] S. Hoyer, J. Sohl-Dickstein, S. Greydanus, Neural reparameterization improves structural optimization, arXiv preprint arXiv:1909.04240 (2019).

[26] A. Chandrasekhar, K. Suresh, TOuNN: topology optimization using neural networks, Structural and Multidisciplinary Optimization 63 (3) (2021) 1135–1149.

[27] L. Piegl, W. Tiller, The NURBS book, Springer Science & Business Media, 2012.

[28] M. Abramowitz, Handbook of mathematical functions with formulas, graphs, and mathematical tables, nbs, Applied Math. Series 55 (1964) 887.

[29] CAD source: modified KVLCC2 and modified pyiges, https://github.com/fedetamburlin/cad-pinn-souces (2026).

[30] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, A. Courville, On the spectral bias of neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 5301–5310.

[31] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[32] D. R. S. Saputro, P. Widyaningsih, Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method for the parameter estimation on geographically weighted ordinal logistic regression model (GWOLR), in: AIP conference proceedings, Vol. 1868 (1), AIP Publishing LLC, 2017, p. 040009.