Transition from Adjoint Level Set Topology to Shape Optimization for 2D Fluid Mechanics

J.R.L Koch[a,*], E.M. Papoutsis-Kiachagias[a], K.C. Giannakoglou[a]

[a] *National Technical University of Athens (NTUA), School of Mechanical Engineering, Parallel CFD & Optimization Unit*
Athens, Greece

Abstract

Most optimization problems in the field of fluid mechanics can be classified as either topology or shape optimization. Although topology and shape have been considered mutually exclusive optimization methods since their inception, it is conceivable that they will find choicest solutions in tandem, with shape optimization refining a solution found by topology. However, linking the topology optimization problem to that of shape is not trivial and, to the authors' knowledge, has yet to be formally attempted. This paper proposes a novel transitional procedure that post-processes 2D adjoint topology solutions, fitting the interface between the solid and fluid topological domains to create a parameterized solution which can be used as either a CAD-compatible representation of the interface or a source for grid generation from which a shape optimization loop can be initialized. The interface to be fit can be extracted from any topological field with distinct fluid and solid domains, meaning that the proposed transition process is independent of the topology approach utilized. To conveniently describe the interface between the solid and fluid topological domains, the topology optimization process employed in this paper is filtered using the level set method. The interface is fit with non-uniform rational B-spline (NURBS) curves through application of sensitivities garnered from the solution of an auxiliary inverse design problem which aims at reducing the difference between signed-distance fields generated about both the NURBS curve being optimized and the section of interface being fit. The geometry defined by the fit NURBS curves is then (optionally) used to build a boundary-fitted grid on which a shape optimization loop is performed. The parameterized result of the topology to shape transition process is compared to that of shape optimization in 2D cases with internal, incompressible fluid flows.

Keywords: Topology-to-Shape Transition, CAD-compatibility, Constrained Topology Optimization, Level Set Method, Shape Optimization, Adjoint-Based Optimization

*Corresponding author
Email addresses:* `kochjamesr@gmail.com` (J.R.L Koch), `vaggelisp@gmail.com` (E.M. Papoutsis-Kiachagias), `kgianna@central.ntua.gr` (K.C. Giannakoglou)

1. Introduction

First proposed in [5] for the design of structural mechanics, topology optimization (TopO) has been expanded to the field of fluid mechanics through the introduction of a blockage term (β) into the Stokes [2, 6, 11] and laminar [10], turbulent [20] steady and unsteady, [15, 8], flow equations. This blockage term acts as a design variable for each grid cell and its corresponding field is controlled to minimize an objective function. The interested reader can find an overview of relevant TopO methods, focusing mainly on structural but including also multi-disciplinary applications, in [7, 26]. Shape optimization (ShpO), on the other hand, was initially introduced as a shape control technique to reduce an objective function, [24], and has been expanded to fluid mechanics by altering the original boundary shape and, thus, its flow solution.

Both the TopO and ShpO techniques require the ability to affect the current solution of a problem through alteration of design variables such that the provided objective function(s) decrease or increase. To achieve this, the TopO and ShpO processes in this paper employ the continuous adjoint method to supply objective function sensitivities, thereby driving the ensemble of design variables to its optimal value-set. Continuous adjoint has been extensively developed for both laminar and turbulent flows, [24, 13, 4, 21, 22], and is applied as the driving method for the TopO and ShpO processes used in this work.

TopO is commonly employed in fluid mechanics to develop optimal internal-flow paths between prescribed inlets and outlets according to provided objective function(s), but itself produces solutions which are not CAD-compatible. Alternatively, ShpO may only alter boundaries of known connectivity within a geometry to search for an optimal solution, but uses far fewer design variables than TopO (*i.e.*, the boundary nodal or control point coordinates instead of internal field values) and is inherently CAD-compatible if the parameters of the CAD model are used as the design variables. Two avenues of research thus present themselves: firstly, to be able to process a TopO solution accurately such that it becomes CAD-compatible and, secondly, to take that solution and use it to initialize a ShpO process, thereby enabling comparison between TopO and ShpO. To properly initialize a ShpO solution, information pertaining to TopO's internal-flow boundaries (the 'interface' between fluid and solid domains) is required. For porosity-based TopO, such an interface can be found by using the so-called projection methods, [12, 25, 28], which tend to produce (nearly) discrete porosity values or through post-processing the porosity field by filtering it to binary values, [14]. In this work, the level set (LS) method, [3, 9, 29], is used to define the interface of the two domains, which is computed using the continuous adjoint method. First proposed in [19], the LS method is a conceptual framework which introduces a LS field, effectively a signed-distance field, through which moving interfaces can be mapped. To integrate the LS method into TopO, β must become a function of the LS field: the solution domain is modeled as solidified where the LS field is negative, fluidized when positive and zero at the interface. To progress the interface toward the optimal solution, the cell-centered LS field is updated utilizing adjoint-based sensitivities and corrected through application of an accurate, fast-marching signed-distance

correction algorithm. A review on LS methods for (structural) optimization can be found in [27].

The TopO-to-ShpO transition process initially subdivides the TopO solution's interface into segments such that each segment pertains to a connection between an inlet and outlet. These interface segments are used to construct individual signed-distance fields which act as target solutions to an inverse design problem (referred to as the 'field matching' problem) which tries to iteratively fit a non-uniform rational B-spline (NURBS) curve to each segment. The NURBS curve's control points are computed by minimizing the difference between a narrow band (NB) signed-distance field built about each NURBS curve and the curve's corresponding 'target' signed-distance field. This minimization problem is simplified when using the LS-based TopO as the tools required for generating signed-distance fields are already in place. However, provided that similar tools are present in porosity-based approaches, the proposed method could be extended for that framework as well. A boundary-fitted grid is then generated from the fit NURBS curves and used to initialize an adjoint-driven ShpO process, generating a parameterized optimal solution.

This paper applies the proposed method for conjoining the TopO and ShpO processes to three, 2D internal-flow cases with the goal of minimizing volume-averaged total pressure losses between the inlet(s) and outlet(s). The first is an example case used to walk the reader through the TopO-to-ShpO transition process. The second explores the benefits of stopping TopO before convergence and using the transition process to allow ShpO to find the optimal solution. The third pertains to investigating the fitting capabilities of the transition process, *i.e.* what limitations it faces and how accurately it can fit a provided TopO interface. A fluid volume percentage constraint is used to generate a more tractable TopO solution in the first two cases. All cases and in-house code pertaining to the coupling process are implemented within OpenFOAM 2.2.1 [1].

2. Topology Optimization

2.1. Flow Equations

The modeled fluid flow is governed by the steady-state Navier-Stokes equations for laminar, incompressible fluid flows. The blockage formulation of TopO requires the addition of a term which contains a function of the blockage variable ($0 \leq \beta \leq 1$) into the momentum equation [20]. The resulting flow (state) equations are

$$R_p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{1a}$$

$$R_{v_i} = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] + \underbrace{f(\beta)v_i}_{Blockage\ Term} = 0 \tag{1b}$$

where $f(\beta) = \beta \cdot \beta_{MAX}$, β_{MAX} is a large, positive user-defined value and repeated indices indicate Einstein summation, which will apply to all equations found in this Section. For this work, β_{MAX} is defined through

the ratio of viscous and inertia forces present in each case (i.e. the Darcy number) in the manner defined in [17]. As is seen in Section 2.3, the prescribed limits of β are enforced through enhancing TopO with the level set method, thereby making β dependent on the local LS field variable ϕ through a Heaviside relation. If $f(\beta)$ becomes comparatively large, the added term becomes dominant, forcing the velocity toward zero and effectively rendering the domain solid. Conversely, if this term is comparatively small, the flow is uninhibited and the domain remains fluid [6, 11]. The usual boundary conditions associated with internal aerodynamics (prescribed inlet velocity, constant outlet pressure and no-slip condition along the walls) are imposed for all cases discussed in this paper.

2.2. Adjoint Equations, Constraints, Boundary Conditions and Sensitivities

In order to properly discuss the formulation of the adjoint equations, the basic TopO optimization problem must be examined. Without loss of generality, the objective function J to be minimized is that of the volume-averaged total pressure losses between the inlet(s), S_I, and outlet(s), S_O,

$$J = -\int_{S_I} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS - \int_{S_O} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS \tag{2}$$

where n_i are the components of the outward (from the fluid to boundary) normal vector.

To account for the inclusion of an equality (volume) constraint $C = 0$ into the TopO process, a Lagrangian function L is introduced, according to the Augmented Lagrange Multiplier (ALM) method [16]. Thus, the (optionally constrained) optimization problem to be considered is

$$\min \quad L = J - \lambda C + \omega C^2 \tag{3}$$

where λ and ω are the Lagrangian multiplier and a dynamically changing weighting scalar, respectively [16]. If the constraint is not imposed, $L \equiv J$. As the constraint applied in this paper is volumetric, its contribution to the field sensitivities $\frac{\delta L}{\delta \beta_m}$ is directly computed using the LS field variable. Thus, it is discussed after the LS formulation of TopO in Section 2.4 for clarity.

To formulate the adjoint field equations and boundary conditions, the Lagrangian is further augmented by the flow equation residuals as follows

$$L_{aug} = L + \int_{\Omega} q R_p d\Omega + \int_{\Omega} u_i R_{v_i} d\Omega \tag{4}$$

with q and u_i being the adjoint pressure and adjoint velocity components, respectively. After lengthy derivation, [14], the variation of eq. (4) w.r.t. the β design variable set becomes

$$\frac{\delta L_{aug}}{\delta \beta_m} = \int_{\Omega} R_q \frac{\partial p}{\partial \beta_m} d\Omega \quad + \int_{\Omega} R_{u_i} \frac{\partial v_i}{\partial \beta_m} d\Omega \quad + \int_{\Omega} v_i u_i \frac{\partial f(\beta)}{\partial \beta_m} d\Omega$$
$$+ \int_{S} \mathcal{BC}_1 \frac{\partial p}{\partial \beta_m} dS + \int_{S} \mathcal{BC}_{2,i} \frac{\partial v_i}{\partial \beta_m} dS - \int_{S} \frac{\partial}{\partial \beta_m} \left[\nu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] u_i n_j dS \tag{5}$$

where m is the computational cell index and S is the boundary of the computational domain Ω. For simplicity, L_{aug} will be referred to as L for the remainder of this work. The R terms in eq. (5) become the adjoint field equations by eliminating all terms with field integrals containing partial derivative(s) of the flow variables w.r.t. the design variables by setting their multipliers against zero. It should be noted that these partial derivatives are technically total derivatives: $\frac{\delta()}{\delta\beta_m} \equiv \frac{\partial()}{\partial\beta_m}$ for TopO since the computational grid remains unchanged [14].

The adjoint continuity and momentum field equations which make $\frac{\delta L}{\delta\beta_m}$ independent of $\frac{\partial p}{\partial\beta_m}$ and $\frac{\partial v_i}{\partial\beta_m}$ are

$$R_q = \frac{\partial u_j}{\partial x_j} = 0 \tag{6a}$$

$$R_{u_i} = -v_j\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{\partial q}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \underbrace{f(\beta)u_i}_{Blockage\ Term} = 0 \tag{6b}$$

The BC terms in eq. (5) are related to the adjoint boundary conditions in that they describe conditions to be met in order for $\frac{\delta L}{\delta\beta_m}$ to become independent of partial derivative(s) of the flow variables w.r.t. the design variables along the boundary. Thus, the adjoint boundary conditions are found (see [14]) by requiring that conditions

$$\mathcal{BC}_1 = u_i n_i + \frac{\partial J}{\partial p} = 0 \tag{7a}$$

$$u_i t_i = 0 \tag{7b}$$

$$\mathcal{BC}_{2,i} = u_i v_j n_j + \nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)n_j + (u_j v_j - q)n_i + \frac{\partial J}{\partial v_i} = 0 \tag{7c}$$

be met. Eqs. (7a) and (7b) pertain to inlets and wall boundaries, make $\frac{\delta L}{\delta\beta_m}$ independent of $\frac{\partial p}{\partial\beta_m}$ and $\frac{\partial \tau_{ij}}{\partial\beta_m}$ (where τ_{ij} are the stresses) and allow the normal and tangential components $u_i n_i$ and $u_i t_i$ of the boundary adjoint velocity to be computed. Eq. (7c) pertains to outlets and makes $\frac{\delta L}{\delta\beta_m}$ independent of $\frac{\partial v_i}{\partial\beta_m}$. After satisfaction of the field adjoint equations and their boundary conditions, eq. (5), which defines the sensitivity derivative of L w.r.t. the blockage field design variable(s), becomes

$$\frac{\delta L}{\delta\beta_m} = \int_\Omega v_i u_i \frac{\partial f(\beta)}{\partial\beta_m} d\Omega = v_i^m u_i^m \Omega^m \beta_{MAX} \tag{8}$$

where, for $\frac{\delta L}{\delta\beta_m}$, the blockage function $f(\beta)$ is dependent only on the m-th grid cell with volume Ω^m associated with the discrete description of the field domain.

2.3. Level Set Topology Formulation

To allow the TopO interface to be conveniently described for the subsequent transition to ShpO, the level set (LS) method was employed to control β and will be synonymous with the term TopO hereafter. The LS method represents design domains in terms of a signed-distance function ϕ which is differentiable at

the interface Γ separating them. The field of the signed-distance function must have a gradient magnitude of 1 everywhere ($\|\nabla\phi\| = 1$) and defines its physical domains through the relation $\phi_i = \{$negative $\forall\ i \in \Omega_{\text{Solid}},$ zero $\forall\ i \in \Gamma,$ positive $\forall\ i \in \Omega_{\text{Fluid}}\}$, [29]. Thus, Γ is the zero-ϕ isoline of the LS field.

The LS method used in this work is not a 'true' LS method (in which the zero LS contour is propagated using shape sensitivities computed on the interface) as it essentially utilizes what [26] describes as 'ersatz' materials. To move Γ toward the optimal solution, the LS field is subjected to a process of convection such that the total derivative remains zero for the Lagrangian reference system. If $\|\nabla\phi\| = 1$ everywhere and the normal convecting velocity, V_n, of Γ is defined as its unit normal $\frac{\partial\phi}{\partial x_i}$ multiplied by a vectorial quantity V_i, then the LS convection equation is

$$\frac{d\phi}{dt} = \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial x_i}\frac{\partial x_i}{\partial t} = \frac{\partial\phi}{\partial t} + V_i\frac{\partial\phi}{\partial x_i} = \frac{\partial\phi}{\partial t} + V_n = 0 \tag{9}$$

The discrete form of eq. (9),

$$\phi_m^{i+1} = \phi_m^i - (\Delta t)V_{n_m} \tag{10}$$

practically stands for a steepest-descent step for the ϕ field, with Δt being the discrete convection's step size, m a cell index and superscript i the iteration index. V_n is comprised of two different contributions: those from the TopO sensitivities derived from the objective function J, V_{n_J}, and those from the constraint C (if included), V_{n_C}. Thus, the discrete convection velocity V_{n_m} is defined as

$$V_{n_m} = V_{n_{m\,J}} + V_{n_{m\,C}} = \frac{\delta L}{\delta\phi_m} \tag{11}$$

and discussed in Section 2.4.

After eq. 10 is explicitly applied to Γ, $\|\nabla\phi\| = 1$ does not necessarily hold, requiring that the ϕ field undergo a signed-distance correction procedure (SDCorr) after each convection application. The procedure employed in this paper corrects $\forall\phi_m \in \Gamma$ using the existing ϕ values for the cells holding Γ and then conducts a fast marching method to solve for the remainder of the ϕ domain [18]. If the computational domain is large, performing this fast marching method for the entire ϕ domain can be expensive. Thus, a narrow-band (NB) technique is employed: the ϕ values are subject to SDCorr only within the NB. Although this means that the SDCorr must be performed after each convection step (Γ moves, requiring the NB cell set to be reassessed), the overall computational cost of the procedure becomes negligible if the NB is small.

The LS field can be initialized from any existing β field through the relation

$$\phi_m = \phi(\beta_m) = 1 - 2\beta_m \tag{12}$$

which defines a starting zero-ϕ isoline, and therefore Γ, as the β interface. Thus, Γ is synonymous with the β interface and will replace it for the remainder of this paper. It should be noted that although Γ is essentially the design boundary of TopO, and will effectively match the initial design boundary of ShpO

(which is part of the ShpO grid boundary) later on, Γ is not to be confused with the general denotation for the boundaries of a computational grid, S.

Eq. (12) is a one-way expression: once the LS field has been convected to generate a new, hopefully better, Γ, an inverse application of this equation will not give an accurate β field: after convection, the LS field must undergo a SDCorr, resulting in values which may not respect the prerequisite $0 < \beta < 1$ bounds if eq. (12) is used. Thus, after initialization, the dependence between the LS field and β is reversed in order to respect these bounds: β is set by ϕ according to the continuously differentiable, piecewise NB sigmoidal Heaviside relationship

$$\beta_m = \beta(\phi_m) = H(\phi_m) = \begin{cases} 1 & \phi_m < -d_{NB} \\ \frac{1}{2}\left[1 - \frac{\phi_m}{d_{NB}} - \frac{1}{\pi}sin\left(\frac{\pi\phi_m}{d_{NB}}\right)\right] & |\phi_m| \leq +d_{NB} \\ 0 & \phi_m > +d_{NB} \end{cases} \tag{13}$$

where $d_{NB} > 0$ is the user-defined normal distance from the zero-ϕ isoline (in either direction) which sets the limits of the NB. In order to ensure stable convection, d_{NB} is limited to include a single cell on either side of Γ. This Heaviside function differentiates into a finite and continuous function $D(\phi_m)$, $i.e.$,

$$\frac{\delta\beta_n}{\delta\phi_m} = \frac{\delta H(\phi_n)}{\delta\phi_m} = D(\phi_m)\delta_m^n \tag{14}$$

where δ_m^n is a Kronecker delta. This relationship is used in Section 2.4 via application of the chain rule to eq. (13) to recast the objective function's TopO adjoint sensitivities into TopO convecting velocities.

2.4. Level Set Convecting Velocity Term Formulation

The LS convection described by eq. (10) requires the set of normal convecting velocities V_{n_m} defined in eq. (11). Expanding eq. (11) gives the individual contributions of the objective function and constraint,

$$V_{n_m} = \frac{\delta L}{\delta\phi_m} = \frac{\partial L}{\partial\beta_l}\frac{\delta\beta_l}{\delta\phi_m} + \frac{\partial L}{\partial C}\frac{\delta C}{\delta\phi_m} \tag{15}$$

Without inclusion of the volume constraint, the TopO sensitivities can be recast as LS sensitivities by exploiting eq. (14), rendering the first velocity component in eq. (11) as

$$V_{n_{m J}} = \frac{\partial L}{\partial\beta_m}D(\phi_m) \tag{16}$$

Likewise, by holding the β field constant w.r.t. the LS field variable and taking the derivative of eq. (3) w.r.t. said field variable, the contribution of the constraint function C to the convecting velocity is

$$V_{n_{m C}} = (-\lambda + 2\omega C)\frac{\delta C}{\delta\phi_m} \tag{17}$$

where $\frac{\delta C}{\delta\phi_m}$ is found through analytical differentiation of the constraint function. The volume constraint applied in this paper requires a certain percentage of Ω to be Ω_{Fluid} (assessed through a total, volume-averaged sum of the β field). From a production perspective, defining the allowed fluid volume through such

7

a constraint can help meet material usage restrictions. The constraint function is defined as

$$C = \frac{1}{2}(\mathcal{V}_{Fluid} - \mathcal{V}_{Fluid,Tar})^2, \quad \mathcal{V}_{Fluid} = \frac{\int_\Omega (1 - H(\phi))d\Omega}{\int_\Omega d\Omega} \qquad (18)$$

where $\mathcal{V}_{Fluid,Tar}$ is the user-defined percentage of Ω_{Fluid} to be maintained within Ω.

2.5. Level Set Process Overview

The LS formulation of the TopO process (within each TopO iteration) can be summarized as follows:

0. Initialize ϕ from an existing β field via eq. (12).

1. Solve eqs. (1) to get the flow solution.

2. Solve eqs. (6) to get the adjoint solution.

3. Compute $D(\phi_m)$ and the LS convecting scalar velocities V_{n_m} via eqs. (11), (16) and (17).

4. Progress Γ toward the optimal solution by updating ϕ using eq. (10).

5. Conduct the SDCorr process and get new, correct ϕ values within the NB of the newly convected Γ.

6. Update the β field via eq. (13). Set all cells owned by walls to 'solid' (if specified). Set all cells owned by inlets and outlets to 'fluid' to ensure continuity. Return to Step 1.

3. 2D Topology-to-Shape Transition (TtoST)

The process of ShpO implemented in this work takes parameterized (CAD-compatible) curves or surfaces as inputs in order to build a computational grid and generate a refined, possibly improved, parameterized solution. In the present work, these inputs are comprised of non-uniform rational B-spline, or NURBS, curves. The mathematical relevance of NURBS curves to the present work is briefly discussed in Section 3.5: further explanation and theory can be found in [23]. The goal of the proposed transition process is to automatically generate and fit, as closely and efficiently as possible, NURBS curves to TopO solutions so as to have an initial boundary which matches the TopO geometry and can be fed directly into a grid-generation program to begin the ShpO process. Alternatively, the solution of the transition process could be used as a manufacturable solution to TopO without considering a continuation to ShpO. The transitional fitting algorithm requires a known set of 'target' points, which, for this paper, is defined as the Γ solution of the TopO formulation described in Section 2.3.

The procedure for the proposed TtoST process is divided into four main steps: target isoline (i.e. Γ) segmentation, NURBS curve initializations, 'target' signed-distance field generation and fit refinement via a 'field matching' algorithm. The first step divides the Γ information contained in TopO's ϕ solution into portions which pertain to each relevant inlet-outlet connection. These portions of Γ become the target curves to be individually fit by a set of NURBS curves. The second step initializes each NURBS curve based on a least squares fitting problem using a predefined degree and number of control points. As this fit may be

poor, a search for a more desirable degree and number of control points can be optionally conducted. The third step generates a target solution for the inverse design problem solved in the fourth TtoST step, which attempts to minimize the difference between Γ-based and NURBS curve-based signed-distance fields. Each Γ connection separated in Step 1 is treated as an independent Γ from which a signed-distance field is built throughout the entire computational domain through the SDCorr process used to correct TopO's convected Γ in Section 2.3. The fourth and final step of the TtoST process is known as the field matching algorithm: by generating a signed-distance field about the NURBS curves initialized in Step 2 (again, using the SDCorr process) and comparing its values to the target signed-distance field generated in Step 3, coordinates for the NURBS control points are updated such that the total difference between the two fields is minimized. The four steps of the TtoST process are explained through their application to a representative test case, with the results being discussed in Section 3.6.

For all TopO and ShpO cases in this paper, the boundary of the computational grid is denoted as S, with the subscripts I, O, W, FW and Y designating inlet, outlet, wall, 'fixed' wall (walls not pertaining to the design space of a case: see case descriptions) and symmetry-plane portions of the boundary, respectively. For all TopO cases discussed herein, the β field is initialized to zero ('all fluid' initialization where the interface begins at the walls) since, based on authors' experience, this approach makes the LS-based TopO less susceptible to entrapment in local minima.

3.1. TtoST Explanatory Test Case: 2D Single Inlet-Double Outlet Duct

The case used to exemplify the TtoST process is a laminar, 2D case with one inlet and two outlets, each of which is connected to the main design domain by 'fixed' wall (S_{FW}) entrance channels. The design domain, i.e. the portion of the grid where β is allowed to be altered, excludes these entrance channels. The mesh is Cartesian with 13000 cells, 10000 of which are in the design domain. The uniform velocity at the inlet at $1.0\frac{m}{s}\hat{x}_1$ ($\hat{x}_i$ is the unit vector along the ith Cartesian axis), and the kinematic viscosity ν of the fluid is $0.001\frac{m^2}{s}$. A d_{NB} of $0.01m$ was selected for the LS process and the fluid volume constraint was specified with $\mathcal{V}_{Fluid,Tar} = 0.3$ within the design domain, leading to an overall target volume of $\mathcal{V}_{Fluid,Tar} = 0.462$ when the entrance channels are taken into account. The objective function J is given by eq. (2). The geometric data and ϕ field solution (found by the TopO process) of the example case are shown in fig. 1. The case is initialized with the internal field of β specified as fluid ($\beta = 0$) and cells adjacent to the design walls (S_W) specified as solid ($\beta = 1$), and thus initially holding Γ. The progression of Γ throughout the TopO process can be seen in fig. 2.

3.2. TtoST Step1: Target Isoline Segmentation

In Section 2.3 the term 'isoline' was used to refer to the zero-ϕ contour of the LS field (*i.e.* Γ): this term will be re-cast w.r.t. the TtoST process for clarity. For the remainder of this section, isoline will refer to
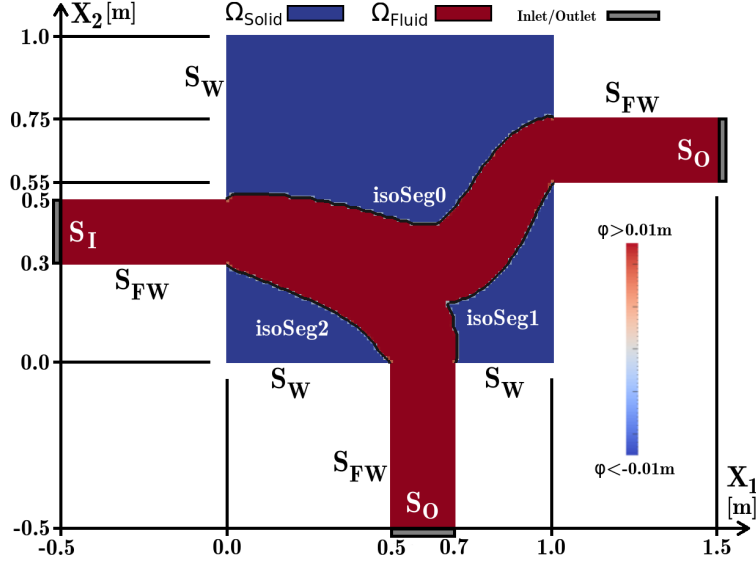
Figure 1: The TopO minimal volume-averaged total pressure loss ϕ solution and geometry of the TtoST example case. The Γ of the solution can be seen as a black line between the fluid (red) and solid (blue) topological domains.

the set of computational cells which hold the TopO solution's Γ. This cell set can be thought of as pairs of computational cells which contain ϕ values with different signs. The faces between these cells across which ϕ switches sign are referred to as 'isoFaces'. In order to obtain a target curve against which a NURBS curve's fitting is to be compared, Γ must be separated into segments, as each wall (inlet-outlet boundary connection S_W) of ShpO should be represented by a single NURBS curve. Thus, the isoline must also be decomposed into distinct portions referred to as 'isoSegs' (the sets of computational cells holding these Γ segments). The TtoST process targets the segmentation of the isoline rather than Γ, as Γ is defined by the information contained within the isoline's cells.

For 2D geometries, the isoline must be separated into as many segments as there are NURBS curve end-point pairs to be connected: half the number of existing end-points if topological islands are not present. The NURBS curve's end-points are defined at the centers of the edges shared by boundary face pairs (2D OpenFOAM meshes are actually 3D meshes with one cell in $\hat{x}_3$ for which the flow and adjoint equations are solved in $\hat{x}_1$ and $\hat{x}_2$ only) which have a face in both the S_W set and one of the following sets: S_I, S_O or S_{FW}. The normal vector at each end-point (used to define the initial, positive normal direction for a NURBS curve starting there) is defined by interpolating the gradient of ϕ in the cells owned by its pair of boundary faces. For the explanatory case there are six end-points, requiring the isoline to be broken into three isoSegs. The isoSegs are generated by interpreting a field which is 1 for isoline cells and 0 otherwise and tracking the isoline cells between end-points. During the tracking process, points which lie on Γ are gathered so as to have an ordered set of points against which to fit NURBS curves in Section 3.3. Thus,
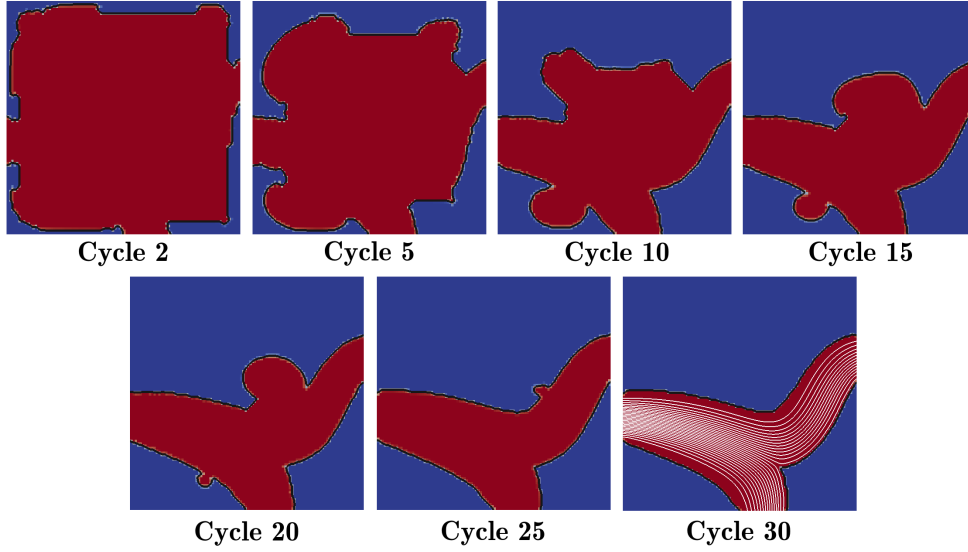
Figure 2: The progression of the TtoST example case's Γ (shown as a black line) for the optimization cycles 2, 5, 10, 15, 20 and 25 preceding the conclusion of the TopO process at cycle 30 (shown with streamlines).

these 'isoPts' sets effectively map Γ. The process for generating the isoSeg and their isoPts is outlined below, with the specific path chosen by it for the example case being depicted in fig. 3.

1.1 From the first end-point, march along S_W until the isoline is found. For each S_W-owned cell passed, add its cell center to the isoSeg's isoPts list.

1.2 Find the face between the isoline cells which are owned by S_W and across which the sign of ϕ changes. This is the starting isoFace for the face marching process.

1.3 *UNTIL* a new isoFace owned by two boundary cells is found,

(a) Find the isoPt of the current isoFace through linear interpolation of ϕ and add it to the isoPts.

(b) Loop over the face edges.

(c) *IF* there is a face across which ϕ switches sign and has not been considered yet, that face becomes the next isoFace.

1.4 March along S_W starting in the positive ϕ direction until a known end-point is reached. For each cell add its cell center to the isoSeg's isoPts list.

1.5 *IF* all isoSegs are generated, *BREAK*.

ELSE start from an end-point that has not yet been considered.

3.3. TtoST Step 2: NURBS Curve Least Squares and Parameter Search Initialization

The final step of the TtoST process (described in Section 3.5) consists of solving an inverse design problem for each of Γ's isoSeg that iteratively fits a NURBS curve to it. As these NURBS curves do not yet
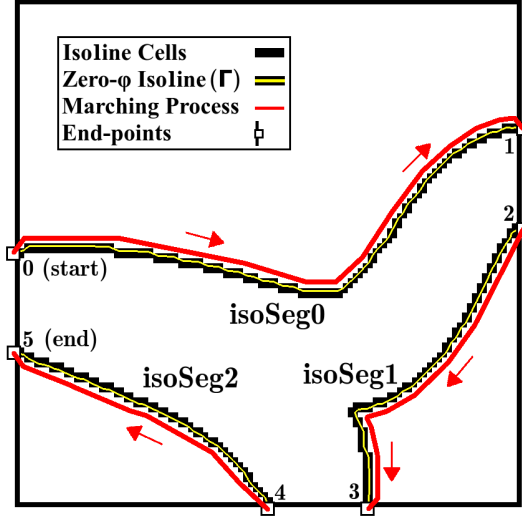
Figure 3: NURBS curve end-points are denoted as numbered squares on the case boundary. Γ is contained between pairs of the black isoline cells. The marching segmentation process starts with end-point 0 and marches until it connects to end-point 1, generating isoPts along isoSeg0 and saving the Γ cells belonging to it. The next end-point available is 2, which is similarly connected to 3 (isoSeg1). Finally, end-point 4 is connected to 5 (isoSeg2). The process terminates because no unused end-points remain.

exist, Step 2 of the TtoST process initializes them using the end-point, starting normal and ordered isoPts list information defined in Step 1. First, P_I points are generated on each NURBS curve I, where P_I is the number of isoPts for the Ith isoSeg to be fit, and a least squares fitting problem is solved. The least squares problem is supplied with a user-defined number of control points (nCPs) and degree which are identical for all NURBS curves to be fit; 8 control points and degree 4 for the considered example case. Performed with fixed initial nCPs and degree, the least squares fitting can be poor if portions of the isoSeg have high curvature (isoSeg1) or change concavity several times (isoSegs 0 and 1): see fig. 3 for a reference to isoSeg labeling. These fitting issues are exemplified in fig. 4 and discussed in Section 3.6.

To mitigate the effect of these matching errors, a parameterization search can be performed for each NURBS curve individually after its initialization via least squares, potentially altering the nCPs and degree of the NURBS to decrease the error of the least squares fit. This algorithm is defined as follows:

2.1 Using the end-points, build the NURBS basis. Generate P_I points on the NURBS curve. Perform a least squares fitting with the current nCPs and degree to generate the control points.

2.2 Perform Step 2.1 with incremented (++) nCPs : IF the least squares error decrease exceeds a defined percentage, return to Step 2.1 using the new nCPs.

2.3 Perform Step 2.1 with incremented (++) degree ($UNLESS$ at degree nCPs-1) and decremented (- -) degree ($UNLESS$ at a user-defined lower limit OR degree 2): IF the least squares error decreases for
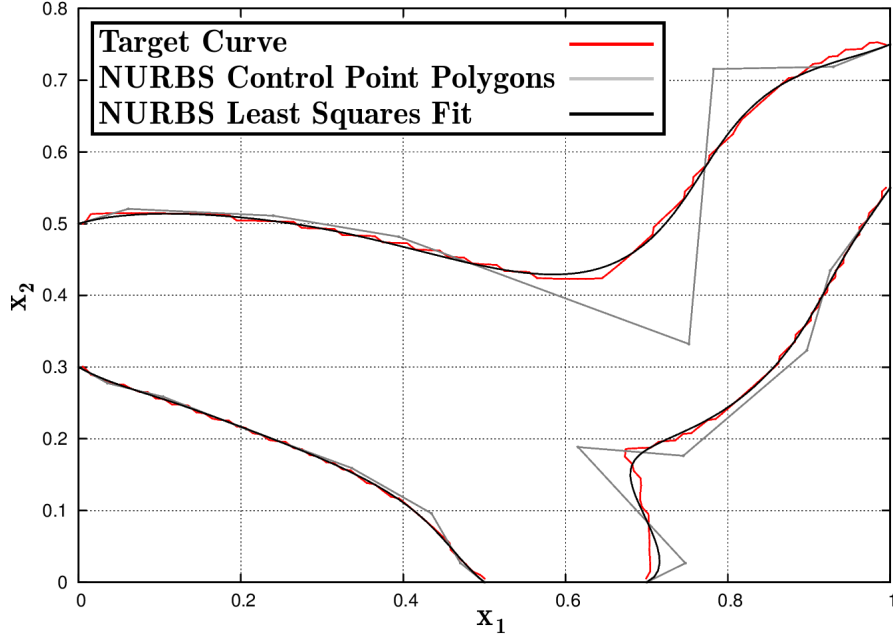
12

Figure 4: The TtoST NURBS curves and their control point polygons plotted against the example case's Γ solution after their initialization via least squares fitting using 8 control points and degree 4 for each isoSeg.

the incremented or decremented degree, build the NURBS curve again with that degree via Step 2.1 and repeat Step 2.3.

2.4 Build the NURBS curve with the new nCPs and degree.

The new nCPs and degree found for each NURBS curve of the example case using this algorithm can be seen in Table 1. A plot of the resulting NURBS curves, their control point polygons and the isoPts after this parameter search process can be seen in fig. 5. For the example case, the fitting found by the parameter search is quite good, as the least squares error-decrease limits of control point addition were quite lax. For stricter addition limits or more complicated geometries, least squares based fitting is unlikely to give a high degree of fitting accuracy. To ensure an accurate fitting, the TtoST process subsequently solves the inverse design problem discussed in Section 3.5.

NURBS#	New nCPs	New Degree
0	10	3
1	10	4
2	9	3

Table 1: NURBS parameters after the parameter search algorithm for the example case. The numbering of each NURBS curve matches that of the isoSegs (see fig. 3).
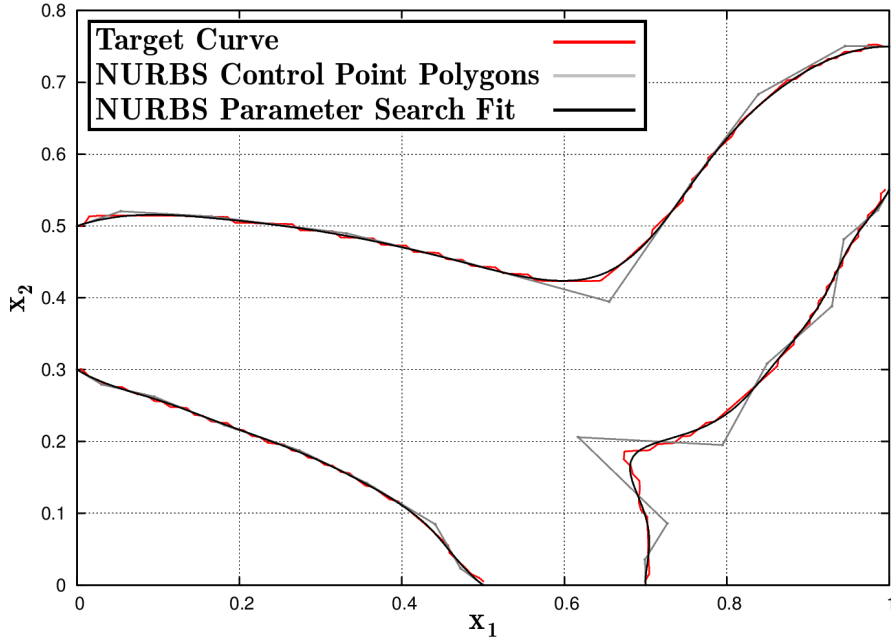
13

Figure 5: The TtoST NURBS curves and their control point polygons plotted against the example case's Γ solution after the parameter search algorithm. See Table 1 for a listing of the nCPs and degree for each curve which the parameter search used to decrease the least squares error of the NURBS curve's fit.

3.4. TtoST Step 3: Signed-Distance Target Field Generation

As mentioned in Section 3.5, the final step of the TtoST process (described in Section 3.5) consists of solving a 'field matching' problem for each of Γ's isoSeg that iteratively fits a NURBS curve to it. Signed-distance fields must be generated for both the NURBS curve and the isoSeg it is attempting to fit, since comparison between the two quantifies the accuracy of the NURBS curve's fit. Using separate isoSeg signed-distance fields for each fitting process in this manner means that each NURBS curve only sees the isoSeg it is attempting to fit and will not consider any of the others. The generation of these signed-distance fields is conducted using a SDCorr process similar to that used in Section 2.3. The signed-distance field of each NURBS curve, ϕ_N, is constructed within a NB similar to that of the LS process' 2–cell NB calculation, but to a larger, user-defined normal distance d_{NB} from the isoSeg. ϕ_N is constructed for each iteration of the field matching process in Step 4 after the NURBS curve is moved toward the optimal fit, to be discussed further in Section 3.5. The signed-distance field of each isoSeg is the 'target' field which ϕ_N desires to (locally) emulate and is thus referred to as ϕ_{Tar}. As each NURBS curve, and thus ϕ_N, can exist anywhere within Ω and the cells within its NB must have ϕ_{Tar} values for comparison, ϕ_{Tar} must exist for all Ω, making the d_{NB} of its SDCorr process effectively the distance between Γ and the furthest point on the case's boundary from it. The ϕ_{Tar} fields for each isoSeg within the example case are shown in fig. 6.
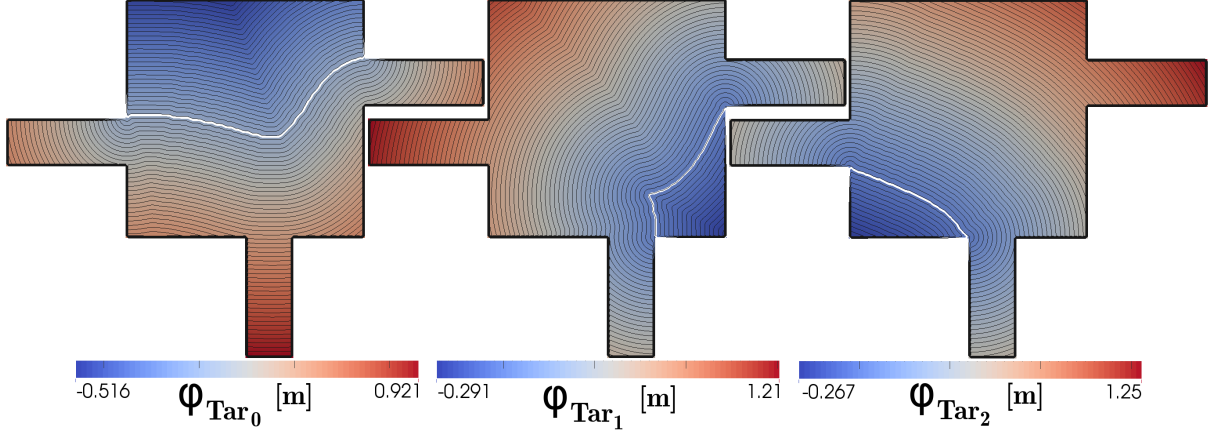
14

Figure 6: The ϕ_{Tar} fields of each isoSeg to be used in the TtoST field matching algorithm (see Section 3.5).

3.5. TtoST Step 4: NURBS Curve Field Matching Inverse Design Problem

As briefly discussed in Section 3.4, the final step in the TtoST process is an iterative field comparison between each NURBS curve's NB and target signed-distance fields ϕ_N and ϕ_{Tar}. The fitting process attempts to minimize $\Delta\phi_{NB}$, the aggregate, absolute-value difference between the two fields within the NURBS curve's NB cell set, through control point, and thus NB, displacement. Since the field matching process desires to find the optimal location of each NURBS curve's set of control points, its design variables are the degrees of freedom (DOFs) of those control points. A control point's displacement is dependent on sensitivities which quantify how the current $\Delta\phi_{NB}$ relationship would change if the control point were to be moved along its DOFs. Thus, to properly explain the field matching algorithm, the formulation of its sensitivities must be discussed. The field matching objective function J_{FM} to be minimized and its sensitivity w.r.t. a design variable b_i are

$$J_{FM} = \frac{1}{2} \int_{\Omega} H_2(\phi_N)(\phi_N - \phi_{Tar})^2 d\Omega \tag{19}$$

$$\frac{\delta J_{FM}}{\delta b_i} = \int_{\Omega} \left[\frac{1}{2} \frac{\partial H_2(\phi_N)}{\partial \phi_N}(\phi_N - \phi_{Tar})^2 + H_2(\phi_N)(\phi_N - \phi_{Tar}) \right] \frac{\delta \phi_N}{\delta b_i} d\Omega \tag{20}$$

and $H_2(\phi_N)$ is the following double Heaviside function (which allows only the NB to be considered for integration)

$$H_2(\phi_N) = \left[\left(1 + e^{-\epsilon(\phi_N + d_{NB})}\right)\left(1 + e^{\epsilon(\phi_N - d_{NB})}\right) \right]^{-1}, \quad \epsilon = min\left(\frac{5}{(\text{cell}\Delta X)}, \frac{100}{d_{NB}} \right) \tag{21}$$

where cellΔX is a user-defined grid parameter (*i.e.* the width of a computational cell for a uniform mesh) which was defined as $0.01m$ for the example case. The d_{NB} chosen for eq. (21) in the example case was $0.1m$. Eq. (21)'s inclusion in eq. (19) means that only those cells within the field matching process' NB need be considered for control point displacement sensitivity calculations. Both it and its relevant derivative can

15

be seen in fig. 7. The first term in the integrand of eq. (20) describes how altering the shape of a NURBS curve through control point displacement affects displacement of the NB, while the second term describes the change of $\Delta\phi_{NB}$ within the current NB affected by control point displacement.
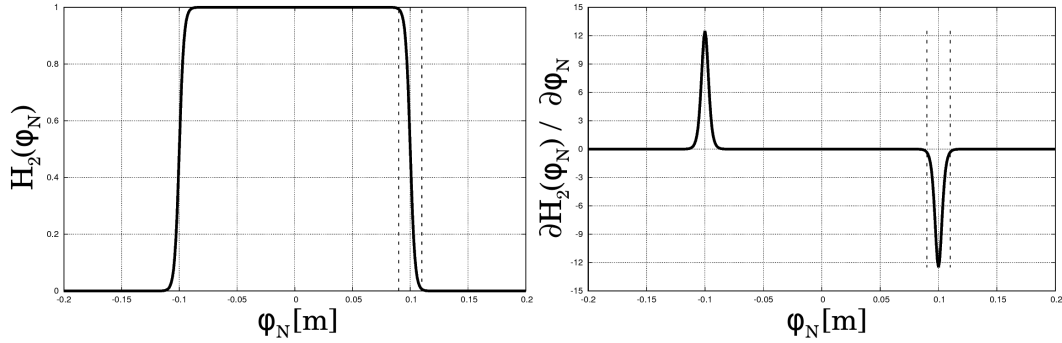

Figure 7: A plot of eq. (21) (left) and its derivative (right) for the parameters used in the example case: cell$\Delta X = 0.01m$ and $d_{NB} = 0.1m$. The broken lines enclose the width of a single cell, *i.e.* cellΔX.

For the sake of the sensitivity derivation which follows, in which repeated indices do not indicate Einstein summation, it is helpful to recast the design variables so that the control point and particular DOF the design variable pertains to are easily recognized. If q is index of a control point within the total control point set of size χ which design variable b_i pertains to and n is the component of control point q which b_i represents, then $b_i = B_q^n$ (*i.e.* $B_q^0 = x_{1_q}$ and $B_q^1 = x_{2_q}$) . A NURBS curve has a total set of points P that will be broken into subsets P_r which pertain to the points affected by a design variable B_r^k (the switch of q and n with r and k is intentionally done to distinguish arbitrarily considered design variables from the indexing a NURBS curve's CP set requires). If N is the set of basis functions of the NURBS defined by the parametric variable u, within each subset P_r, points are defined as

$$p_j^k = \sum_{r=0}^{\chi-1} N_r(u_j) B_r^k \tag{22}$$

where j is the index of the point within P_r and k is the point's component ($p_j^0 = x_{1_j}$, $p_j^1 = x_{2_j}$). Each point p_j has a set $M_{r,j}$ of cells m within the NB which it 'owns' (*i.e.*, those for which the SDCorr process defines ϕ_m based on the distance from cell m's center c_m to p_j). Here, m is the cell index of a cell's vectorial center c or signed-distance ϕ value and is effectively irrelevant outside of the NURBS's NB for sensitivity considerations due to the inclusion of the H_2 function within the sensitivities. If B_r moves, the set P_r of these NURBS points which are affected by it will also be displaced, requiring a sensitivity relationship between B_r and $p_j \in P_r$ and between $p_j \in P_r$ and $c_m \in M_{r,j}$ This coupling relationship is exemplified in fig. 8.

The term within eq. (20) that requires a coupling from a design variable to a computational cell is $\frac{\delta\phi_N}{\delta b_i}$,

16
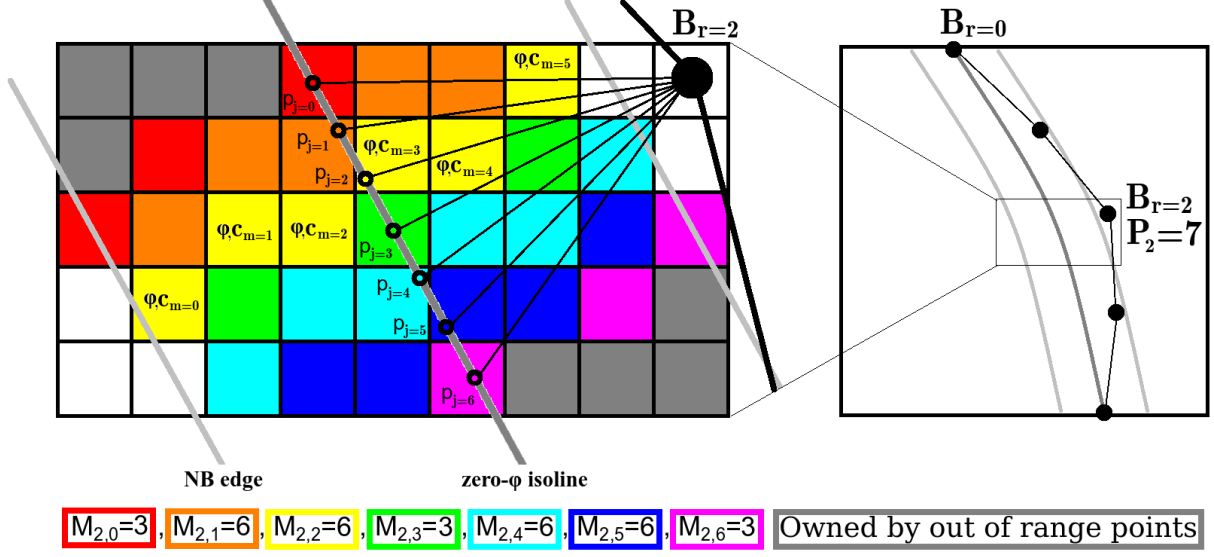
Figure 8: An illustrative example of the ownership relationship between a control point B_r, the points p_j of the set P_r on the NURBS curve which it affects and the cells of sets $M_{r,j}$ which are dependent on those points as defined by the SDCorr process used to generate ϕ_N within the NURBS NB. The sensitivity relationship between B_r and a cell m within the NB (eq. (27)) is non-zero for the design variables of control point 2 for colored cells only (note that sets $M_{r=2,j=0,2,5,6}$ would likely include more cells if the picture were not truncated). Grey cells within the NB would have their ϕ_m (i.e. ϕ_N) values determined by points not shown in the illustration during the SDCorr process' construction of ϕ_N, i.e. 'out of range' points. The cells of $M_{2,2}$ have been further labeled as an example of how iteration over $p_{j=2}$'s cell set could occur.

which is recast as $\frac{\delta\phi_m}{\delta B_q^n}$ and derived as follows

$$\frac{\delta\phi_m}{\delta B_q^n} = \sum_{j=0}^{P_q-1} \sum_{k=0}^{D-1} \frac{\delta\phi_m}{\delta p_j^k} \frac{\delta p_j^k}{\delta B_q^n} \tag{23}$$

where D is the number of relevant DOFs for the given geometry ($D=2$ for the 2D cases presented in this paper). The relationship between a point on the NURBS curve and its owning design variable is

$$\frac{\delta p_j^k}{\delta B_q^n} = \sum_{r=0}^{\chi-1} N_r(u_j) \frac{\delta B_r^k}{\delta B_q^n} = \sum_{r=0}^{\chi-1} N_r(u_j)\delta_q^r \delta_n^k = N_q(u_j)\delta_n^k \tag{24}$$

Plugging eq. (24) into eq. (23),

$$\frac{\delta\phi_m}{\delta B_q^n} = \sum_{j=0}^{P_q-1} \sum_{k=0}^{D-1} \frac{\delta\phi_m}{\delta p_j^k} N_q(u_j)\delta_n^k = \sum_{j=0}^{P_q-1} \frac{\delta\phi_m}{\delta p_j^n} N_q(u_j) \tag{25}$$

where ϕ_m is taken to be the distance between its owning NURBS point and cell m's vectorial center c_m,

$$\phi_m^2 = (p_l - c_{m(l)})^2$$
$$\frac{\delta\phi_m}{\delta p_j^n} = \frac{p_l^n - c_{m(l)}^n}{\phi_m}\delta_j^l \tag{26}$$

17

where l is an index of P_q and subscript $m(l)$ denotes that a cell of index m is owned by the point with index l. Plugging eq. (26) into (25), we find

$$\frac{\delta \phi_m}{\delta B_q^n} = \sum_{j=0}^{P_q-1} \frac{p_l^n - c_{m(l)}^n}{\phi_m} \delta_j^l N_q(u_j) = \frac{p_j^n - c_{m(j)}^n}{\phi_m} N_q(u_j) \tag{27}$$

with $N_q(u_j)$ effectively evaluating the NURBS's derivative contribution at point p_j. The computation of the field matching control point sensitivities therefore requires a summation over all cells m in the NB, adding each's sensitivity contributions to all relevant control points according to eq. (27).

The field matching algorithm, initialized with the NURBS curves found by the parameter search (see Section 3.3), can now be summarized as follows for each NURBS curve to be fit:

4.1 Generate points on the NURBS curve and perform the SDCorr for a defined d_{NB}.

4.2 Compute J_{FM} via summation of the difference between the ϕ_N and ϕ_{Tar} fields within the NB.

4.3 Calculate sensitivities via eq. (20) and move the control points accordingly.

4.4 IF $\frac{|J_{FM}^n - J_{FM}^{n-1}|}{|J_{FM}^{n-1}|} < \Delta J_{FM_{MIN}}$, $BREAK$.
$ELSE$ return to Step 1.

The field matching algorithm was run for the example case with $\Delta J_{FM_{MIN}} = 1\mathrm{e}{-4}$. A plot of the resulting NURBS curves, their control points and the isoPts after the field matching process can be seen in fig. 9.

3.6. TtoST Process Overview

The isoSegs for the example case's Γ are plotted alongside a comparison of the fitting algorithms' results in fig. 10. IsoSegs 0 and 2 are fit fairly accurately by the least squares fitting used to initialize their NURBS curves. Generally, however, and as is found to be the case for isoSeg1, initialization via this least squares method will not be accurate for isoSegs which have areas of high curvature or traverse large distances and change concavity several times.

If a more accurate NURBS curve initialization is desired, the parameter search must be used. As seen in fig. 10, by adding control points or decreasing the degree, most aspects of isoSegs 0 and 1 are captured by the parameter search. The least squares error limits specified for control point addition and degree alteration within parameter search are important, as for complicated geometries an initialization with a desirably low least squares error will only be achieved through drastically raising the nCPs of the NURBS curves. Additions of large numbers of control points by the parameter search can also be mitigated by either raising the least squares error decrease limit or lowering the maximum nCPs (see Section 3.3).

At first, the field matching process does not seem to play an overly important role in refining the overall fitting of the example case's Γ; as the parameter search gave a good initial fit, it only changed the NURBS curves' control point positions to capture the the area of high curvature in isoSeg1. If the desired result of the TtoST process is to transition to ShpO, one possible interpretation of fig. 10 is that the parameter search
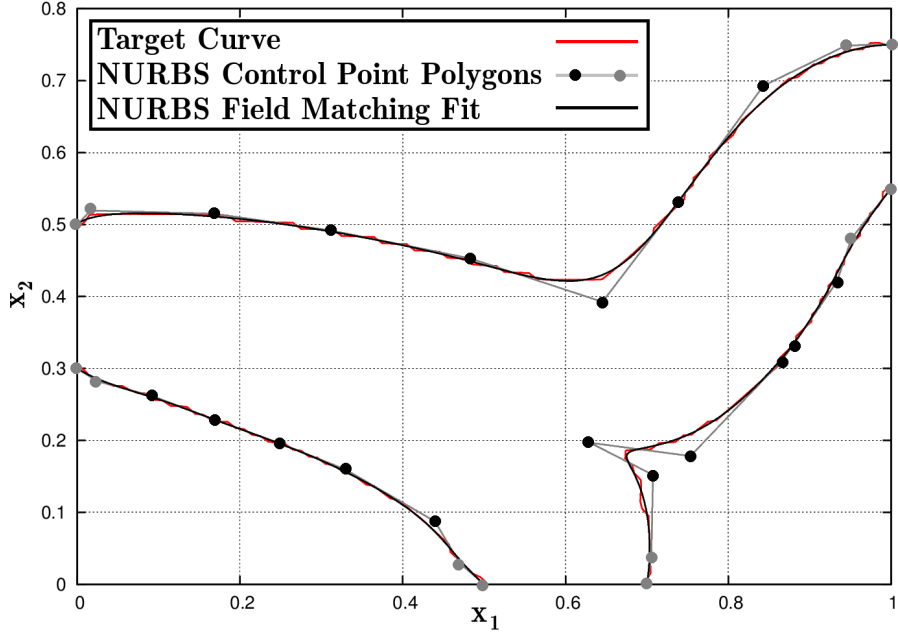
Figure 9: The TtoST NURBS and their control point polygons plotted against the example case's Γ solution after the field matching algorithm. Control points of the control point polygon which are grey were chosen to be fixed during the ShpO initiated from this TtoST solution (see Section 4).

(or even the initial fit provided by least squares) indeed gives an accurate enough approximation of the TopO solution as a ShpO initialization. The validity of this line of reasoning is case dependent as it is related to the resulting design space of ShpO: are there restrictions on how many control points each parameterized wall can have, are minimal control points desired, is the possibility of achieving a good solution more difficult for some ranges of degree and nCPs, does the accuracy of the TopO fitting have a significant effect on solution convergence, etc. See Section 4 for a discussion on the differences between TopO and ShpO solutions. However, if the desired result of the TtoST process is to convert the TopO solution to a CAD-compatible boundary representation, then accurately fitting all aspects of the isoSegs is important, and the alterations made by the field matching process are significant. Again, for more complicated cases the impact the field matching procedure has on accurately mapping Γ increases significantly. Studies quantifying the fitting capabilities of the TtoST process are found in Section 6.

Although the TtoST process has no direct way to prevent the NURBS curves it generates from overlapping, it is inherently designed to be resistant to such a result. Unless Γ of the case is ambiguously close to itself, initializing the NURBS curves with least-squares is usually accurate enough to generate a non-overlapping NURBS set. Even if overlapping were to occur during such curve initialization, field-matching each curve with its own ϕ_{Tar} will accurately refine the accuracy of its fit to the point where overlap is removed.
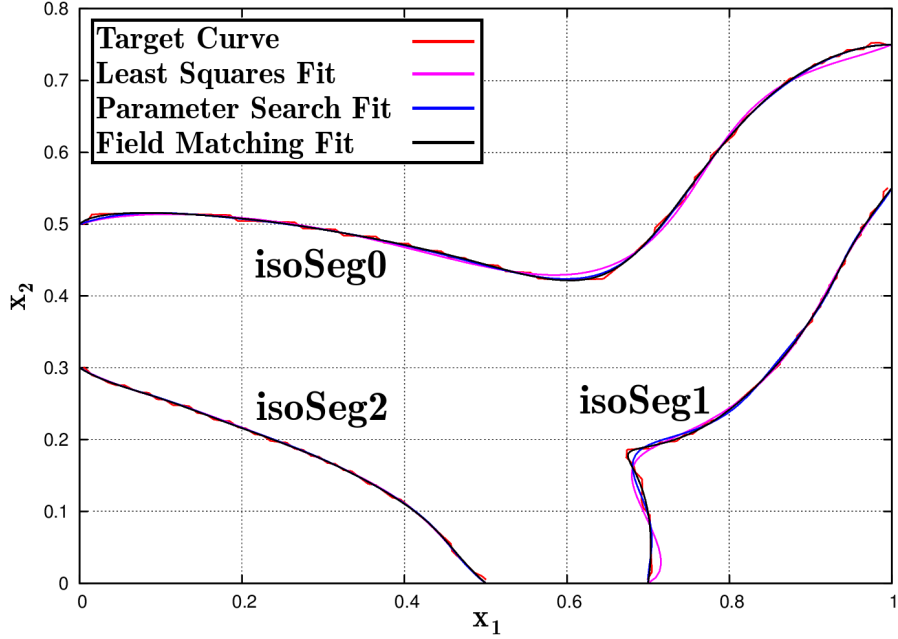
19

Figure 10: The TtoST process' fitting progression.

4. Shape Optimization

The TtoST process (see Section 3) generates a set of NURBS curves which represent the Γ of the TopO solution and which can be treated as both the initial set of solid wall boundaries (S_W) for the primal and adjoint ShpO problem (once a grid has been generated from them) and the parameterized design space of the optimization problem ShpO attempts to solve. The flow and adjoint equations for the ShpO are identical to eqs. (1) and (6) if the blockage terms are excluded. Similarly, the flow and adjoint boundary conditions of the ShpO are identical to those of TopO for inlets, outlets and walls. The design variables b_l for ShpO are the DOFs of the control points which define S_W. The sensitivities for the ShpO control points are computed via

$$\frac{\delta J}{\delta b_l} = -\int_{S_W} \left[\left(\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j - q n_i \right) \frac{\partial v_i}{\partial x_k} \frac{\delta x_k}{\delta b_l} - (q R_p + u_i R_{v_i}) n_k \frac{\delta x_k}{\delta b_l} \right] dS \qquad (28)$$

where $\frac{\delta x_k}{\delta b_l}$ is computed analytically through the NURBS formula. It is beyond the scope of this paper to present the employed continuous adjoint-based ShpO method in further detail; the interested reader will find pertinent material in [22].

When comparing objective functions of the two methods utilized in this work, two things must be taken into account. The first is grid difference: the Cartesian grids used for TopO in this work have a low resolution

(large cell size) compared to the boundary-fitted grids (i.e. with layers of cell-refinement along its boundary) which are generated for the ShpO cases. As Γ is defined by points linearly interpolated between cell centers, the accuracy of its capture is decreased for lower resolutions. This limitation of Γ information leads to the 'jagged' nature of the TopO interface, which in turn leads to pressure losses as the flow is dragged across Γ. The second is inherent to the difference in flow boundaries between the two methods: Γ cannot be considered a true boundary for the flow as there are no boundary conditions imposed upon it. Instead, it is a 'semi-viscous' band in which β smoothly increases in Γ's normal direction according to eq. (13). Thus, although as an optimization method TopO has a larger design space compared to ShpO, its flow solution is also less accurate and, therefore, incapable of obtaining solutions which ShpO finds. The effect of these two inconsistencies is that a difference in objective function is generally seen for similar geometries when converting from TopO to ShpO.

ShpO was initialized from the explanatory case's TtoST solution with a boundary-fitted grid of 50000 cells. For all ShpO cases presented in this work, S_I, S_O, S_Y and S_{FW} boundaries are considered fixed during the optimization process. In addition, the end and first control points of each design wall were fixed (denoted as grey dots in fig. 9). After conducting a ShpO, the objective function was reduced by 2% and the target volume was retained. The boundary of ShpO solution is compared to the parameterized TtoST NURBS solution in fig. 11. The ShpO process shows preference for raising the upper and narrowing the lower exit ducts.
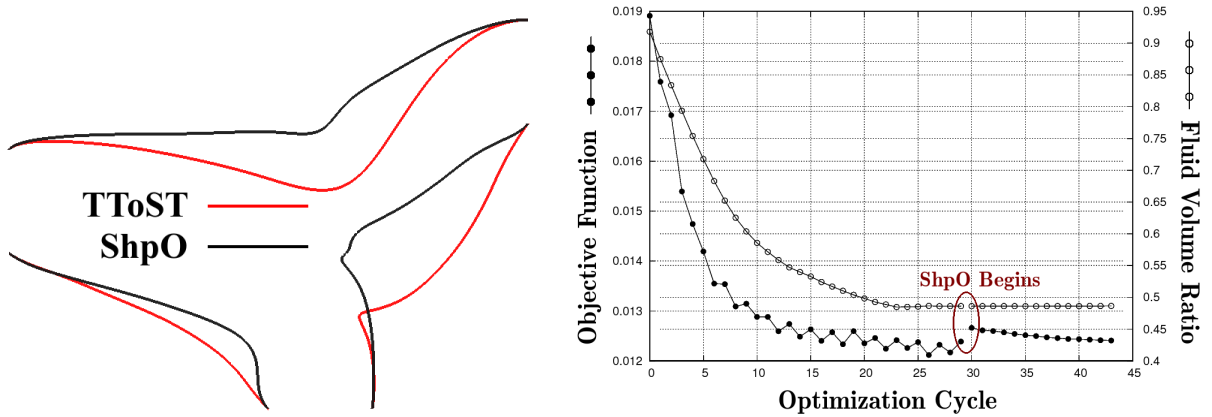

Figure 11: (left): The example case's ShpO boundary: entrance channels not shown. (right): The objective (eq. (2)) and constraint (eq. (18)) functions' progression between TopO to ShpO. ShpO is seen to decrease its objective function by 2% while retaining the prescribed volume ratio.

The trend comparisons of the objective and constraint functions between TopO and ShpO are shown in fig. 11. Due to the imposed volume constraint opposing the expansive sensitivities, TopO developed minor steady-state oscillation, and was thus considered converged at cycle 30. The TopO case was run to convergence for completeness, though in many applications, and as is done in Section 5, TopO would be

21

used simply to initialize a flow channel, then let ShpO converge the solution. This is done since ShpO is considered to be faster than TopO w.r.t. the required number of optimization cycles: Γ can only be displaced within the NB, while the boundaries of ShpO are displaced by updating the location of the design variables controlling them. Thus, the total amount of distance covered by the boundary per optimization cycle is potentially higher.

5. Dual Bottleneck Duct Case: Stopping Topology Prematurely

To explore the possibility of employing the TtoST method developed in this work as a tool for expediting the convergence of an optimization, a more complex 'dualBottleneck' case is presented. The case has a single inlet in the upper left of the case domain, two outlets in close proximity at the bottom right, all of which are connected to the main design domain by 'fixed' wall entrance channels. Two 'bottleneck' locations fixing the area of flow passage are defined between the inlets and outlets. The dualBottleneck has a Cartesian grid of 18200 cells in total, with 15200 being in the design domain, and the same inlet velocity, flow conditions, d_{NB}, objective function and β initialization as the case presented in Section 3. The volume constraint is set to enforce $\mathcal{V}_{Fluid,Tar} = 0.3$ within the design domain, making an overall target volume of $\mathcal{V}_{Fluid,Tar} = 0.415$ when the entrance channels are taken into account. The case geometry, ϕ field solution found by the TopO process, TtoST fitting solution and the final ShpO solution (entrance channels not shown) are found in fig. 12. The progression of Γ throughout the TopO process can be seen in fig. 13. The corresponding listing of TtoST NURBS parameters found using the parameter search algorithm (with the least squares fitting problem's initial nCPs and degree parameters defined as 10 and 5, respectively) are presented in Table 2.

NURBS#	New nCPs	New Degree
0	15	3
1	10	3
2	10	4

Table 2: NURBS parameters after the parameter search algorithm for the dualBottleneck case. The numbering of each NURBS curve matches that of the isoSegs (see fig. 12)

Unlike the example case in Section 3, the dualBottleneck case was prematurely stopped at iteration 15of the TopO process for the reasons discussed at the end of Section 4. Thus, the γ and ω ALM parameters were increased in strength to ensure the constraint be respected by the time of the TopO process' termination. The two bottlenecks present an obstacle to the ShpO process, as displacement of the wall is desired to remain inside the original boundary of the TopO case, even though ShpO does not know where this boundary was. Thus, control points which could affect boundary displacement outside of the original design domain of
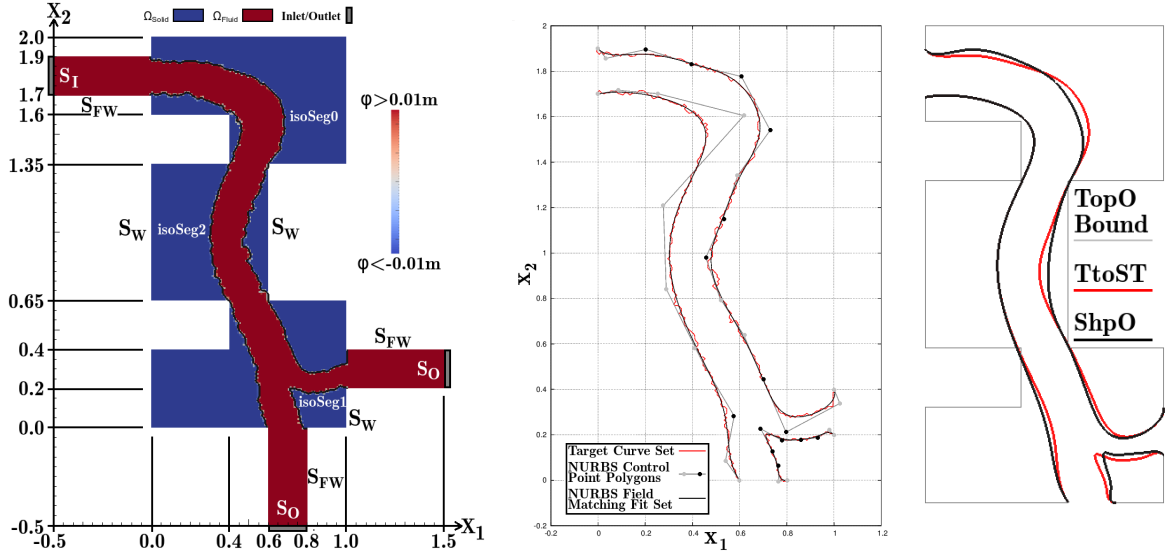
22

Figure 12: dualBottleneck case solution set: (left): Case geometry and TopO ϕ solution with Γ shown as a black line between the fluid (red) and solid (blue) topological domains. (center): TtoST solution. Black and light grey dots indicate which control points were allowed to move and kept fixed, respectively, during the ShpO process. (right): The ShpO boundary solution (entrance channels not shown) overlaid that of the TtoST.

TopO were fixed (marked as grey dots in fig. 12, center). In order to decrease the total pressure loss, the ShpO process (run on a boundary-fitted grid of 25000 cells) enlarges the entrance channel, straightens the channel connecting the two bottlenecks via expansion and expands the lower exit channel. To maintain the volume constraint, the ShpO process narrows the less-sensitive right exit channel and sharpens the entry angle of the into the first bottleneck. As can be seen in fig. 12, the constraints imposed on the ShpO process' control point movements successfully forced the process to respect the original domain of TopO.

The TopO and ShpO objective and constraint functions are plotted in fig. 14. TopO's volume constraint forces an increase in the objective and begins to fight the flow sensitivities by cycle 10, leading to oscillatory behavior. The desired percentage of fluid volume in the design domain is achieved by the time TopO is prematurely ended. Due to the differences between TopO and ShpO discussed in Section 4, a significant objective function decrease is seen when transitioning between the two. This decrease is enhanced due to the fact that TopO has not yet converged and the sensitivities are still quite large (and oscillatory), resulting in a very jagged Γ (see fig. 12, center). The ShpO process decreases the objective function by more than 11% while recovering the prescribed fluid volume percentage.

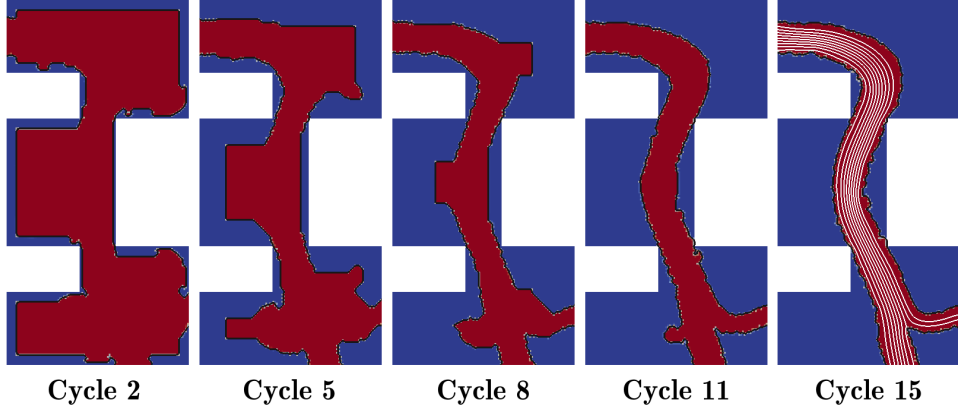| Cycle 2 | Cycle 5 | Cycle 8 | Cycle 11 | Cycle 15 |

Figure 13: The progression of the dualBottleNeck case's Γ (shown as a black line) for the optimization cycles 2, 5, 8 and 11 preceding the 'premature' conclusion of the TopO process at cycle 15 (shown with streamlines).

6. Symmetric Intersecting Flow Duct Case: Investigation of the TtoST Process' Fitting Capabilities

The TtoST process developed in this work allows a TopO solution to be re-defined as a parameterized (CAD-compatible) solution from which a ShpO solution can be initialized. As was touched upon in Section 3.6, the accuracy of the TtoST solution given to initialize ShpO is case dependent and either a simple least squares or accurate field matching fitting of Γ may be desired. However, if the goal of TtoST is to make a TopO solution CAD-compatible, the TtoST process should be able to reliably emulate Γ with a high degree of accuracy. As the TtoST process will always find a better fit when using more control points, quantification of this accuracy can be divided into two requirements: the ability to achieve an accurate fit with few control points which have potentially poor initial locations and the ability to represent the location of the TopO interface with minimal error. Two studies are performed on the case introduced in this section in an attempt to quantify these aspects of the TtoST process.

The case presented in this section is volumetrically-unconstrained and contains two vertically opposing inlets at the top and bottom of the case that create a flow which interacts before exiting from two opposing outlets on the left and right boundaries. The duct is modeled using a $\hat{x}_1$ symmetry plane at $x_2 = 0.5m$. Thus, it is referred to as the 'symmIntersectingFlow' case. The inlets (if reflected across the symmetry plane) and outlet have a width of $0.1m$ and the uniform velocities of the top and bottom inlets are $-0.1\frac{m}{s}\hat{x}_2$ and $0.5\frac{m}{s}\hat{x}_2$, respectively. All inlets and outlets have entrance channels which allow the development of boundary layers. The mesh of the design domain is a Cartesian grid of 6000 cells, with 5000 being in the design domain. The other flow conditions, objective function, d_{NB} and β initialization are the same as in the preceding cases.
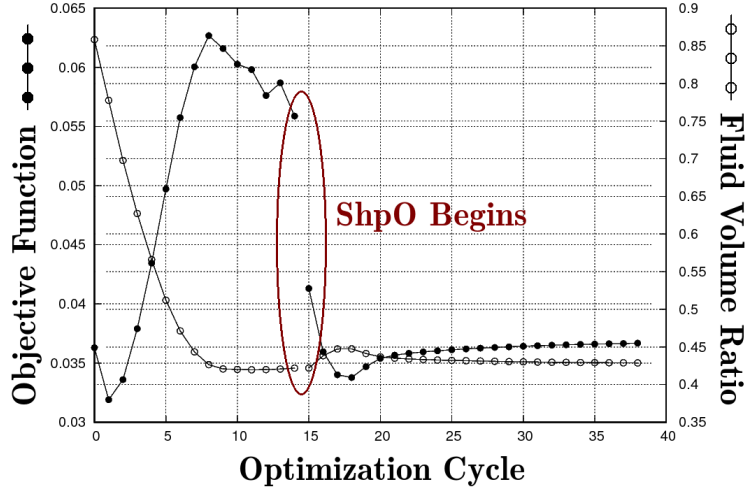
24

Figure 14: The dualBottleneck case's objective function (eq. (2)) and $\mathcal{V}_{Fluid}$ (in eq. (18)) progression from TopO to ShpO. ShpO is seen to decrease its objective function by more than 11%.

In the preceding cases, the parameter search algorithm was used to increase the NURBS curves' initial fitting accuracy of TopO's Γ: from fig. 10, it can be seen that the parameter search finds a much better fitting initialization, which is then easily refined by the field matching process. However, the field matching process should be able to take the potentially poorly-fit NURBS curve initialized using least squares and still obtain a good fit (though the solution of the inverse design problem may take longer). Thus, the fitting solution does not make use of the parameter search process, instead fitting all isoSegs with NURBS defined by 6 control points and degree 4. The convergence criterion for the field matching process discussed in Section 3.5 was set to a strict value of $\Delta J_{FM_{MIN}} = 1e-5$ to ensure an accurate fitting of TopO's Γ. The final ϕ solution of TopO overlaid with the case's dimensions, resulting field matching solution of the TtoST process, and ShpO solution (entrance channels not shown) are found in fig. 15. The progression of Γ throughout the TopO process can be seen in fig. 16.

The TopO process found the optimal solution to be a fluidized asymmetric (in $\hat{x}_2$) cross (in $\hat{x}_1$ symmetry) with a comparatively wider lower entrance channel; a result which is not surprising given that the ratio of imposed velocities between the lower and upper inlets is $5:1$. The strict fitting conditions used for the TtoST process allowed it to achieve a precise fit despite the lack of a parameter search during NURBS initialization. ShpO (run on a boundary-fitted grid of 35000 cells) chose to expand the boundaries of the outlet and upper inlet channels, decreasing the total pressure losses by more than 1.5% by doing so. A plot of the symmIntersectingFlow case's objective function can be seen in fig. 17 for both TopO and ShpO processes. Although the TopO process was again stopped prematurely, the resulting trend is smoother than the dualBottleNeck case since no volume constraint was present to fight flow channel expansion.
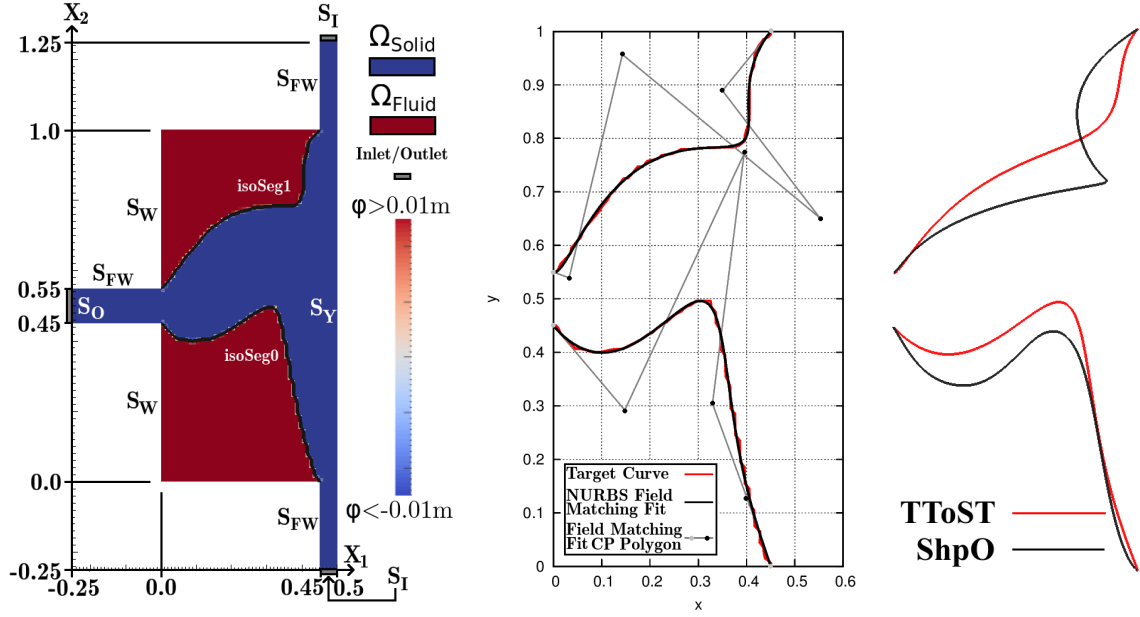
25

Figure 15: symmIntersectingFlow case solution set: (left): Case geometry and TopO ϕ solution with Γ shown as a black line between the fluid (red) and solid (blue) topological domains.. (center): TtoST solution: only the end control points were chosen to be fixed during the ShpO process. (right): The ShpO boundary solution (entrance channels of the ShpO case not shown) overlaid that of the TtoST.

To test if the TtoST process can still obtain an accurate fit with few control points, a study was performed in which the symmIntersectingFlow's Γ is fit using the same parameters as earlier, but with a progressively decreasing number of control points and the maximum allowable degree for that number of control points. The cuves' initialization via least squares and resulting field matching solution for each of the cases are found in fig. 18, which shows that for a reasonable number of control points, the field matching process is capable of accurately fitting Γ. Indeed, the fitting only failed to capture major aspects of Γ if two control points were given as a design space for fitting each isoSeg. Thus, the process is not limited by the control points being used for fitting: control point number becomes an issue of ShpO design space and will be specific to the problem being examined.

A fitting study to assess the accuracy of TtoST process' representation of Γ was also performed. To quantify the accuracy of the TtoST fitting, the following iterative re-fitting procedure was applied to the symmIntersectingFlow Γ's signed-distance field for 10 cycles. First, the current Γ is fit using the TtoST process. Then a signed-distance field is reconstructed from the NURBS curves found by the TtoST process throughout the design domain (i.e. entry channels are excluded). One interpretation of this signed-distance reconstruction is that ShpO is being converted to a TopO solution. This new signed-distance field contains the new Γ to be fit by TtoST. It should be noted that this re-fitting procedure (*i.e.* switching between ShpO and TopO) is not being suggested as a general tool by the authors: by iterating in this manner, areas
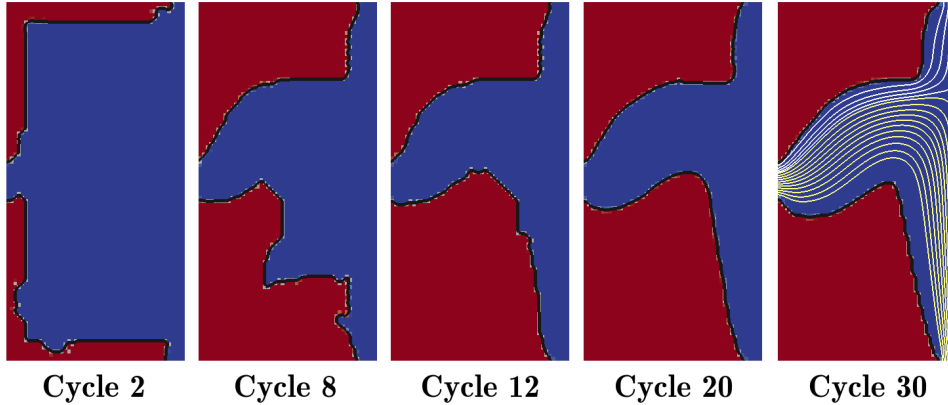
Cycle 2 **Cycle 8** **Cycle 12** **Cycle 20** **Cycle 30**

Figure 16: The progression of the dualBottleNeck case's Γ (shown as a black line) for the optimization cycles 2, 8, 12 and 20 preceding the conclusion of the TopO process at cycle 30 (shown with streamlines).

where fitting is difficult for the TtoST process will accrue minor errors to the point where they are easily identifiable through signed-distance field comparison. To quantify such error between signed-distance fields ϕ, the following metrics are used:

$$E^i_{\phi,Orig} = \frac{\int_\Omega (\phi^i - \phi^0)^2 d\Omega}{\int_\Omega d\Omega} \ , \ E^i_{\phi,Prev} = \frac{\int_\Omega (\phi^i - \phi^{i-1})^2 d\Omega}{\int_\Omega d\Omega} \tag{29}$$

where i is the iteration number of the fitting error progression and ϕ^0 is the signed-distance field generated from the original TopO Γ. For the 10 iterations of the error-comparison process, the values of eq. (29) are graphed in fig. 19.

Although the asymptotic behavior of the error progression is promising, ideally $E^i_{\phi,Orig}$ should be constant and $E^i_{\phi,Orig}$ should be zero after the first iteration, when the jaggedness of Γ is smoothed out. To get a sense of where the progressive error is coming from, the Γ of TopO (iteration 0) was plotted against those of iterations 1 and 10. A field with values of the difference between the signed-distance fields of TopO and iteration 10 was also generated. These correlations can be seen in fig. 20. These comparisons show that most of the error comes from a smoothing of the sharp corner in the middle of the lower isoSeg: the TtoST process finds fitting sharp corners to be more difficult and will only find a good approximation. Once the corner was rounded after several iterations, TtoST was able to fit it with increasing accuracy, generating the asymptotic trends seen in fig. 19. The problem the field matching algorithm has with corners is due to the fact that as the fitting of the NURBS curves becomes more accurate, only continuing to alter the NURBS in problematic areas, the algorithm may see a ΔJ_{FM} (eq. (19)) which is very small, thereby causing the fitting process to desist.
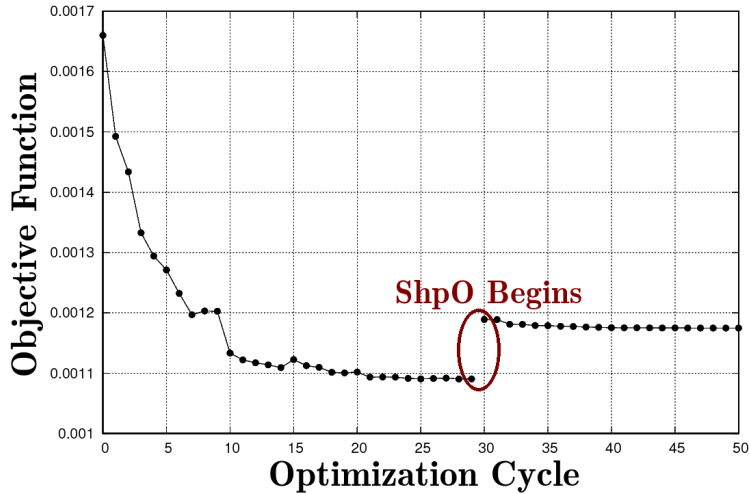
27

Figure 17: The symmIntersectingFlow case's objective function (eq. (2)) progression from TopO to ShpO. ShpO is seen to decrease its objective function by more than 1.5%.

7. Extension of the TtoST Process to 3D

The TtoST process presented in this work can be broken into three main goals: analyze a 2D mesh to get target information to initialize NURBS curves, initialize those NURBS curves, optimize the position of the NURBS curve's control points via the field matching process. When considering extension of the TtoST process to 3D, these goals have varying increase in complexity. The first goal of the 3D TtoST process, that of generating target information, becomes quite complex as the target information will be used to generate NURBS surfaces. The target points will be a surface mesh used for initializing NURBS surfaces. The tracking and subdivision of a surface mesh in an automated manner is a complex problem and will pose the greatest roadblock when extending the TtoST process to 3D. The second goal of the 3D TtoST would be initializing NURBS surfaces. Although this is already a defined science with many options (skinning, etc.), additional fitting considerations must be considered as the resulting set of NURBS surface will be used to generate a parameterized 3D mesh with which to initiate ShpO. For parameterization to be well defined, NURBS surface connectivity should be known and G^1 or G^2 connectivity may be desired alongside the obvious G^0. Such considerations turn NURBS surface initialization into a complex problem. The last goal of the TtoST process, that of field matching, is trivially extended to 3D: the method of field matching presented in this work is 3D inherently.

8. Conclusions

The novel TtoST method for interpreting level set-augmented topology optimization (TopO) solutions was developed in this paper and was shown to be capable of accurately transforming TopO solutions into
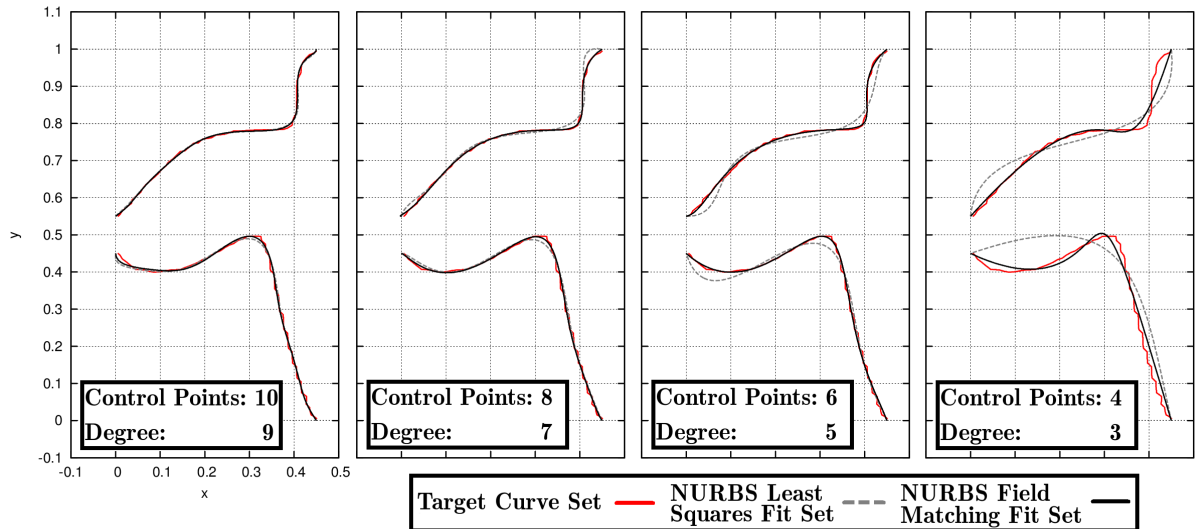
28

Figure 18: Assessment of the TtoST field matching process' limitations w.r.t. a varying number of provided control points. NURBS were initialized using only the least squares fitting problem. The degree of each NURBS curves is the maximum value: one less than the number of control points being used. The field matching algorithm only failed to capture major aspects of TopO's Γ for an initialization with four control points (*i.e.* only allowing two control points to be responsible for fitting each isoSeg).

CAD-compatible geometries. In addition, the TtoST method can be used as a vehicle by which a transition from TopO to shape optimization (ShpO) can be performed by using the result of TtoST as ShpO's initial geometry. The TtoST process is also decoupled from the method which drives TopO and ShpO toward their optimal solutions, which was continuous adjoint for all optimizations performed in this work.

The level set method (LS) was coupled with TopO to allow the interface between TopO's solid and fluid topological domains to be defined throughout the optimization process. To save computational expense, the narrow band technique was utilized in the integrated LS-TopO approach. By employing LS in this manner, information required for the TtoST process to initialize inherently exists throughout the TopO simulation and therefore in its optimal solution. A geometric, fluid-volume limiting constraint was employed for several cases and, in general, can be used to help ensure a topological solution which meets manufacturing or space limitations.

The cases shown in this paper demonstrate the versatility and accuracy of the TtoST process: the TtoST was shown to be capable of capturing solutions for both converged and pre-converged TopO problems, allowing the user to potentially switch to ShpO before TopO has found an optimal solution, thereby decreasing the overall computational cost of the linked TopO-ShpO problem. For all presented cases, ShpO allowed for a significant decrease in the objective function w.r.t. the solution found by TopO, with the improvement being more significant the further from convergence the TopO's solution was. The TtoST process was shown
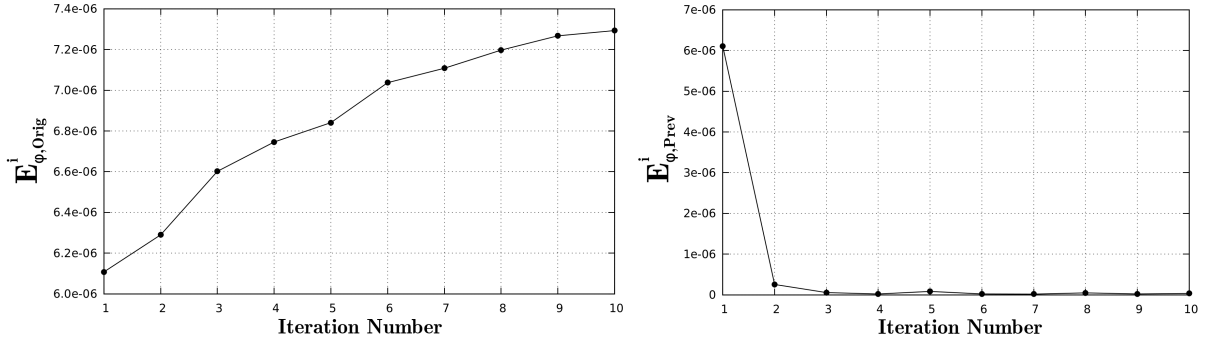
Figure 19: TtoST fitting-error quantification test: Progression of error values defined by eq. (29). The error progression is seen to be stable with the signed-distance field error from the original signed-distance field is asymptotic toward a positive value and the error from the previous iteration being asymptotic toward zero.

to accurately fit the TopO interface, even with few non-uniform rational B-spline (NURBS) curve control points.

A by-product of the work done in this paper is the ability to switch between TopO and ShpO (TopO can be generated from a ShpO solution if ShpO's NURBS curves are treated as the LS interface and a LS field is generated from them) and compare the solutions between the two. Although this is not done extensively in this work, solution comparison is a subject of active research for the authors. Other areas of continuing research include considering persistent islands (those which are part of the TopO solution and which could be implemented into ShpO if the geometry being modeled were expandable to 3D) as viable portions of the TopOs fluid-solid interface to fit, constraining the ShpO process to actively respect the original domain of TopO (this issue was addressed naively in this work by fixing control points which were responsible for the violations), and expanding TtoST to 3D applications.

Acknowledgment(s)

References

[1] www.openfoam.com.

[2] N. Aage, TH. Poulsen, A. Gersborg-Hansen, and O. Sigmund. Topology optimization of large scale Stokes flow problems. *Structural and Multidisciplinary Optimization*, 35:175–180, 2008.

[3] S. Amstutz and H. Andra. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216:573–588, 2006.
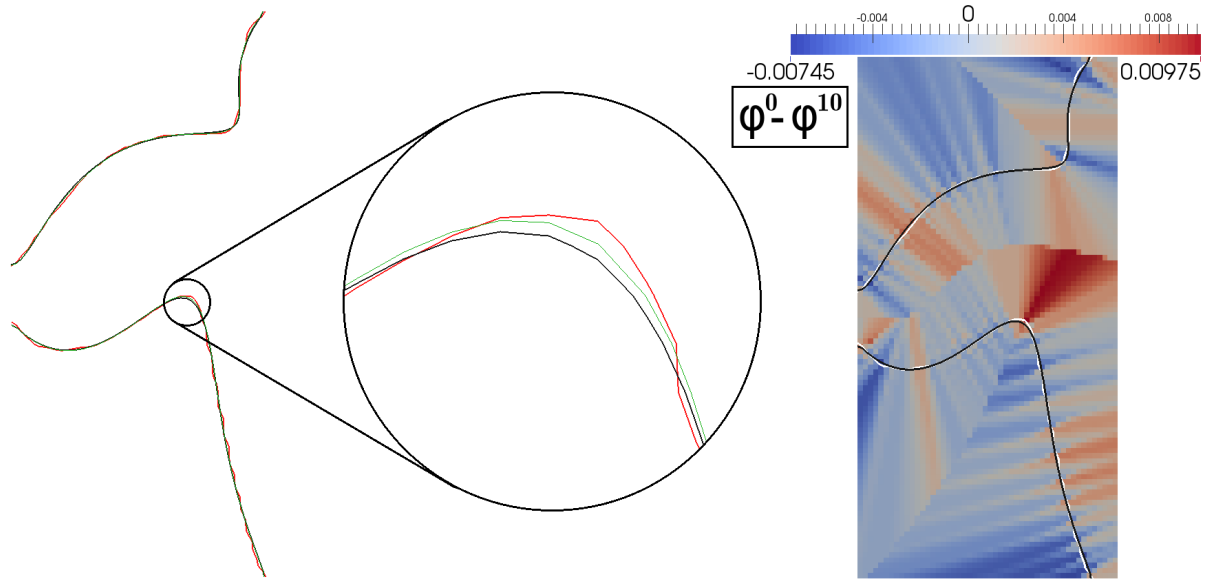
Figure 20: TtoST fitting-error quantification test: (left): Overlaying of the original (iteration 0, red), first (green) and final (black) iteration's Γs, with close-up on the area of significant error. (right): The difference between the original and final iteration's signed-distance fields ϕ. Γs of the original TopO (iteration 0) and final fitting (iteration 10) are shown in white and black, respectively. Minor striations of ϕ difference are due to the smoothing of TopO's Γ which occurs when it is fit by a smooth NURBS curve. The highest source of error can be seen to occur when attempting to fit the corner of the isoSeg which connects the lower inlet and outlet.

[4] WK. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28:443–480, 1999.

[5] M. Bendsoe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Journal of Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[6] T. Borvall and J. Peterson. Topology optimization if fluids in stokes flow. *International Journal for Numerical Methods in Fluids*, 41:77–107, 2003.

[7] JD. Deaton and RV. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization.*, 49(1):1–38, 2013.

[8] Y. Deng, P. Zhang, Y. Liu, Y. Wu, and Z. Liu. Optimization of unsteady incompressible Navier-Stokes flows using variational level set method. *International Journal for Numerical Methods In Fluids*, 71(12):1475–1493, 2013.

[9] XB. Duan, YC. Ma, and R. Zhang. Shape-topology optimization for Navier-Stokes problem using variational level set method. *Journal of Computational and Applied Mathematics*, 222:487–499, 2008.

[10] A. Gersborg-Hansen, O. Sigmund, and R. Haber. Topology optimization of channel flow problems. *Structural and Multidisciplinary Optimization*, 30:181–192, 2005.

[11] JK. Guest and JH. Prevost. Topology optimization of creeping fluid flows using a Darcy-Stokes finite element. *International Journal for Numerical Methods in Engineering*, 66:461–484, 2006.

[12] JK. Guest, JH. Prevost, and T. Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 61(2):238–254, 2004.

[13] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[14] EA. Kontoleontos, EM. Papoutsis-Kiachagias, AS. Zymaris, DI. Paradimitriou, and KC. Giannakoglou. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Engineering Optimization*, 45(8):941–961, 2013.

[15] S. Kreissl, G. Pingen, and K. Maute. Topology optimization for unsteady flow. *International Journal for Numerical Methods In Engineering*, 87(13):1229–1253, 2011.

[16] J. Nocedal and SJ. Wright. *Numerical Optimization.* Springer, 1999.

[17] L. Olesen, F. Okkels, and H. Bruus. A high-level programming-language implementation of topology optimization applied to steady-state Navier-Stokes flow. *International Journal for Numerical Methods in Engineering*, 65:975–1001, 2006.

[18] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

[19] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[20] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids*, 58(8):861–877, 2008.

[21] DI. Papadimitriou and KC. Giannakoglou. Aerodynamic shape optimization using first and second order adjoint and direct approaches. *Archives of Computational Methods in Engineering*, 15(4):447–488, 2008.

[22] EM. Papoutsis-Kiachagias and KC. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.

[23] L. Piegl and W. Tiller. *The NURBS book.* Springer, 1997.

[24] O. Pironneau. *Optimal shape design for elliptic systems.* Springer-Verlag, New York, 1984.

[25] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization.*, 33(4-5):401–424, 2007.

[26] O. Sigmund and K. Maute. Topology optimization approaches. a comparative review. *Structural and Multidisciplinary Optimization.*, 48(6):1031–1055, 2013.

[27] NP. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization.*, 48(3):437–472, 2013.

[28] S. Xu, Y. Cai, and G. Cheng. Volume preserving nonlinear density filter based on Heaviside functions. *Structural and Multidisciplinary Optimization.*, 41:495–505, 2010.

[29] S. Zhou and Q. Li. A variational level set method for the topology optimization of steady-state Navier-Stokes flow. *Journal of Computational Physics*, 227:10178–10195, 2008.