Computers and Structures 319 (2025) 107973

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/cas

An interface-preserving level set update strategy for topology optimisation of mechanical assemblies

Adrian Humphry, Mehran Ebrahimi*, Nigel Morris, Adrian Butscher

Autodesk Research, 661 University Avenue, Toronto, M5G 1M1, Ontario, Canada

ARTICLE INFO

Keywords:
Topology optimisation
Level set method
Constrained Hilbert space extension method
Mechanical assemblies
Geometric interfaces

ABSTRACT

Structural components in assemblies often require specific geometric features — such as cylindrical regions for joints — to function correctly. Standard topology optimisation methods, however, struggle to impose geometric, feature-preserving constraints on selected boundary regions during shape updates. We propose a shape update strategy for level set-based topology optimisation of mechanical assemblies that enables constrained shape updates along user-specified boundaries while allowing free-form updates elsewhere. The constrained regions are limited to affine motions such as translation, rotation, and scaling, providing greater control that is especially valuable in engineering design. This is particularly useful for multi-functional components in larger assemblies, where certain boundaries must retain primitive geometries and vary only within specified limits. For example, when a component must contain a cylindrical aperture to fit a pin of unknown radius, our method allows simultaneous optimisation of the aperture's location, orientation, and size, alongside the component's overall topology. We extend the standard Hilbert space extension method by introducing its constrained variant which incorporates affine motion constraints into the velocity extension. The resulting velocity field satisfies descent direction requirements for the optimisation while ensuring that all feature-preserving constraints are met. We demonstrate the method's effectiveness on canonical structural problems with geometrically constrained boundaries.

1. Introduction

1.1. Topology optimisation of components in an assembly

Topology optimisation is often used to design structural components to optimally fulfil a load-bearing function subject to constraints on material usage, manufacturability, and other factors. The topology optimisation algorithm achieves this by iteratively updating the shape of the component until a measure of its performance under the applied loads (e.g., compliance or average stress) is optimised subject to the given constraints. However, structural components are often not meant to operate in isolation since they are intended to perform their structural function within a larger and more complex engineered system consisting of an *assembly* of interacting components operating together to achieve a range of holistic functions. For example, Fig. 1 shows the structural component known as the *rocker* within a vehicle suspension system, whose purpose is to transmit loads from the wheel sub-assemblies to the chassis through an assembly of rods, springs, dampers, and other components. Therefore, the optimal design of such components should ideally take into account their neighbouring components in the assembly and the assembly as a whole.

The components of an assembly typically interact with their neighbours through well-defined *interfaces* whose nature depends on engineering requirements. For example, interfaces can be welded connections or joint connections that allow only certain relative motions between the components. The vehicle suspension system of Fig. 1 shows such interfaces between the rocker and its neighbours (pull-rods, anti-roll bar, spring/damper sub-assembly). In this system, the components are connected to each other through welds, revolute joints, and cylindrical joints in such a way that the suspension system as a whole achieves a desired dynamical performance appropriate for the operational range of the vehicle. In order to maintain the integrity and function of the assembly, the interface geometries are often highly constrained (e.g., a revolute joint must be cylindrical to permit uni-axial rotational motion, or a weld must be flat for ease of manufacture). Moreover, the optimality of the assembly as a whole can depend critically on the location, orientation, and scale of these interfaces in the design envelope. Therefore,

* Corresponding author.
Email address: mehran.ebrahimi@autodesk.com (M. Ebrahimi).

https://doi.org/10.1016/j.compstruc.2025.107973
Received 2 May 2025; Accepted 26 September 2025
Available online 7 October 2025

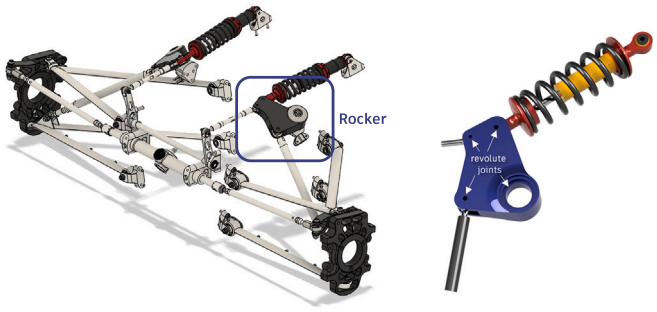


Fig. 1. Vehicle suspension assembly. Left: full assembly. Right: rocker detail with interfaces. *Image courtesy of Andy Harris, Autodesk*.

topology optimisation algorithms suitable for the optimal design of components in an assembly should be able to respect the constraints and design freedoms provided by the interfaces.

In this work, we propose a shape update approach for level set-based topology optimisation to find the optimal shape of components in an assembly while preserving the nature of their interfaces. The essence of our approach is a new *update scheme* for the evolving level set function that allows for the simultaneous free-form optimisation of the non-interface part of component shape boundaries, along with the optimisation of the location, orientation, and scale of the interfaces. We achieve this by *constraining* the local shape updates of the interface regions to rigid motions and orthotropic scalings (which therefore preserve the flat, cylindrical, or spherical regions typical of welds and joints — i.e., the nature of the interfaces) while allowing the non-interface parts of the component shape boundaries to move freely during optimisation. The update can be chosen via a shape sensitivity analysis, which ensures that the combined free-form and constrained displacements of the shape boundary constitute a descent direction for an optimisation objective function. Consequently, for an initial geometry with specified interfaces, if we apply the update in each subsequent iteration of a topology optimisation procedure, the nature of the interface regions will be maintained as the shape and its interface parameters (i.e., location, orientation, and scale) evolve towards optimality.

The level set update scheme that we develop in this work is based on two key modifications of the standard update scheme in level set-based topology optimisation, which is based on solving a Hamilton-Jacobi equation for the level set function of the updated shape using a well-chosen update velocity function. First, we construct an update velocity that is both a descent direction for the optimisation objective function and provides the desired controlled motion of the interfaces at an infinitesimal level. We do this by modifying the *Hilbert space extension* (HSE) method to include constraints in the interface regions. Second, we integrate the velocity via a time-dependent Hamilton-Jacobi equation, which ensures that the macroscopic motion of the level set representing the shape has the desired controlled motion in the interface regions.

In this work, we present our update scheme in a simplified single-component setting that is representative of the full-assembly setting. In this way, we can focus on the details of the update scheme while avoiding additional complexities inherent in working with multiple level set functions, as well as full-assembly simulation and sensitivity analysis.

1.2. Related work

Traditional topology optimisation methods primarily focus on single-component optimisation, applying boundary conditions to fixed regions of the domain that are excluded from the optimisation process. We highlight two key reasons for this: first, these regions often represent interfaces with other existing components in an assembly, and so changes to the shape may interfere with correct mating; second, the topology optimisation algorithm must account for shape sensitivities in these regions, which adds complexity to the process. Despite these challenges, several studies have sought to address such limitations. In [16], the design domain is partitioned such that support elements are optimised on a special thin-edge layer while the material distribution in the rest of the domain is concurrently optimised. Similarly, [22] optimises the positions of interface regions connecting a component to a support structure, alongside the shape and topology of the component represented as a level set. Furthermore, [1] incorporates the position and orientation of support and load regions as design variables within a density-based topology optimisation framework, using a super-Gaussian projection method to streamline the model.

Approaches related to our method of constraining boundary regions to rigid transformations include those that incorporate parametric primitives into continuum-based topology optimisation. In [21], parameterised bars are differentiably projected onto a continuous background grid, where a density-based topology optimisation is used to update the bar parameters. Similarly, [17] co-optimises a beam model fitted to a free-form level set representing a component undergoing topology optimisation, refining both the beam model and the free-form geometry until convergence. In [15], the moving morphable component approach embeds a fixed set of geometric primitives into the design domain, optimising their rotation, scaling, and translation parameters based on sensitivities from a density-based topology optimisation. They rely on representing the entire shape as a collection of geometric primitives, whereas in our work the shapes can be free-form and although we do not explicitly model primitives in our framework, we impose constraints on shape updates to preserve the geometric properties of primitive shapes. This allows, for example, the length and radius of bar-like primitives to be adjusted during the topology optimisation process.

Topology optimisation with geometric constraints has also been studied in the context of optimising for manufacturability, where properties such as hollow tubes or stamped parts [13], minimum thickness [5], and overhang distance [2] are controlled. More recently, a Hilbertian projection approach was proposed in [29] to handle constraints expressible as shape functions (e.g., volume, stress, or microstructural properties) by constructing a constraint-improving descent direction. While these methods also pertain to constrained optimisation, their formulations do not allow for constraining shape updates in specified regions to preserve the geometric nature of an interface, which is the central focus of our work.

Many researchers have also explored topology optimisation of compliant mechanisms, where material placement allows some regions to bend to achieve a desired displacement [8,30]. These mechanisms are typically designed as single-component entities, facilitating continuous representation of both component and joint geometries. However, single-component mechanisms often have a limited range of motion and are less compatible with traditional manufacturing methods compared to multiple-component mechanisms. Additionally, extra care is required to prevent over-stressing the notches designed for bending in compliant mechanisms [23]. In our work, we target a more traditional mechanism design, where multiple components may be combined to form a mechanism with interface regions connecting the components. This problem is also tackled in [6], where joint locations are optimised along with the topology of a set of connected components using the solid isotropic material with penalisation (SIMP) method. They use the adjoint method to solve for the joint location sensitivities and also model a non-design clearance space around the joint geometry. In [27], a Gaussian function is used to parametrise the joint locations connecting multiple components, optimising the joint locations and component shapes with the SIMP method for displacement objectives using non-linear, large-rotation finite element analysis.

In summary, while previous studies have made significant strides in optimising components within assemblies and incorporating parametric primitives in topology optimisation, our level set-based approach offers additional capabilities. Specifically, it enables the optimisation of

the position, orientation, and scaling of any localised region of the geometry, with smooth transitions to the surrounding geometry, ensuring interface/joint compatibility constraints.

1.3. Organisation of the paper

This paper is organised as follows. In Section 2, we first present a brief review of the key steps in the standard sensitivity-based update procedure used in level set-based topology optimisation. We then modify this procedure (specifically, the HSE method for constructing the extension velocity used to update the level set function) to support the optimal design of components of an assembly. Section 3 is the heart of our paper, where we explain how to incorporate constraints on the level set update velocity in the interface regions to ensure that the interfaces are preserved during topology optimisation. This is achieved in two steps: our *constrained Hilbert space extension* (C-HSE) method in Section 3.3, and the velocity integration in Section 3.4. We then apply our approach to solve some representative topology optimisation problems, formulated in Section 4.1, with results presented in Section 4.3. Finally, Section 5 discusses the limitations of the proposed method and outlines directions for future work, and Section 6 concludes the paper.

2. Preliminaries from level set-based topology optimisation

In level set-based topology optimisation, shapes are represented implicitly via *level set functions* defined on a universal design domain $D \subseteq \mathbb{R}^3$ containing all admissible shapes. A typical topology optimisation problem seeks to find an admissible shape, represented by a level set function, that optimises a given objective function and satisfies given constraints (posed as equality and inequality constraints for a collection of constraint violation functions). The level set-based topology optimisation algorithm solves this problem as follows. We start from an initial shape, represented by an initial level set function, that is given *a priori*. The level set function is then updated iteratively in order to improve the objective function value and reduce constraint violations in each iteration until the convergence criteria are met. The specific form of the level set function update in each iteration is determined by performing a sensitivity analysis of the objective and constraint violation functions with respect to shape changes. The update itself is typically performed by solving a so-called *Hamilton-Jacobi equation* for a short time determined by a *line search* with an update velocity that is derived from the sensitivity analysis via a velocity extension procedure.

This work proposes a modification of the standard level set update procedure above in order to support a level set-based topology optimisation approach for the optimal design of multiple components in an assembly. We therefore begin by reviewing the standard update procedure in more detail before turning to the specifics of the modifications we have in mind.

2.1. Shape updates in level set-based topology optimisation

Level set shape representation. Let $F : D \to \mathbb{R}$ be a scalar function defined on the design domain. The sublevel set of value $c \in \mathbb{R}$ is the subset of D defined by $\{x \in D : F(x) \leq c\}$, and the level set of value c is the subset of D defined by $\{x \in D : F(x) = c\}$. When F is sufficiently regular from a mathematical point of view, then these subsets inherit a nice structure. For example, it is a classical result of multivariable calculus that if F is differentiable and c is a regular value, then the c-level set of F is a differentiable, orientable, two-dimensional submanifold of D that bounds the c-sublevel set of F (e.g., [25]). We say that a shape Ω contained in the design domain D is *represented* by the level set function F if Ω is the zero-sublevel set of F.

Level set-based shape updates. Let Ω be a shape represented by the level set function F. A *level set-based shape update* is a one-parameter family of shapes of the form $\Omega_\varepsilon := \{x \in D : F_\varepsilon(x) \leq 0\}$ where $F_\varepsilon : D \to \mathbb{R}$ is a one-parameter family of level set functions that is differentiable with respect to ε. We also assume that ε varies in a small interval containing zero, and $F_0 = F$. Consequently, one can show that Ω_ε is a geometric perturbation of Ω.

Level set-based shape updates via deformations. There are a variety of ways to construct level set-based shape updates. A very geometric way is based on *deformations* of D. A deformation of D is a one-parameter family of invertible mappings $\phi_\varepsilon : D \to \mathbb{R}^3$ that is differentiable with respect to ε, where we assume that ε varies in a small interval containing zero, and ϕ_0 is the identity mapping. Any point $x \in D$ now moves along a trajectory $\varepsilon \mapsto \phi_\varepsilon(x)$ starting at x as a function of the "time" ε. The time-dependent vector field $\Theta_\varepsilon : D \to \mathbb{R}^3$ of velocity vectors of these trajectories is called the *infinitesimal generator* of the deformation because ϕ_ε can be reconstructed from the ordinary differential equation

$$\Theta_\varepsilon(x(\varepsilon)) = \frac{dx}{d\varepsilon}$$

satisfied by the trajectories. Thus, we also say that Θ_ε is the infinitesimal generator of ϕ_ε.

Let Ω be a shape represented by the level set function F, and consider the shape update $\Omega_\varepsilon = \phi_\varepsilon(\Omega)$ for some deformation ϕ_ε. Then, it is quite simple to show that Ω_ε is represented by the level set function

$$F_\varepsilon := F \circ \phi_\varepsilon^{-1}.$$

Thus, every deformation ϕ_ε gives rise to a level set-based shape update F_ε.

Equivalent updates via the transport equation. By differentiating the relationship $F_\varepsilon \circ \phi_\varepsilon = F$ with respect to ε, we have

$$0 = \frac{\partial F_\varepsilon}{\partial \varepsilon} \circ \phi_\varepsilon + \left\langle \nabla F_\varepsilon \circ \phi_\varepsilon, \frac{\partial \phi_\varepsilon}{\partial \varepsilon} \right\rangle.$$

Composing with ϕ_ε^{-1} and using the definition of the infinitesimal generator, we obtain

$$\frac{\partial F_\varepsilon}{\partial \varepsilon} + \langle \nabla F_\varepsilon, \Theta_\varepsilon \rangle = 0. \tag{1a}$$

Together with the initial conditions

$$F_{\varepsilon=0} = F, \tag{1b}$$

we obtain the initial value problem for the level set-based shape update F_ε, called the *transport equation*. It is an example of the class of *Hamilton-Jacobi equations*.

Since the transport Eq. (1) no longer references the original deformation ϕ_ε, the level set update F_ε can be derived directly from the infinitesimal generator Θ_ε simply by solving this equation. We refer to F_ε as the level set-based shape update *generated* by the vector field Θ_ε. It is a fundamental result in the theory of Hamilton-Jacobi equations [14] that this update is equivalent for sufficiently small ε to the deformation-based shape update described in the previous paragraph.

2.2. Choosing the level set-based shape update via sensitivity analysis

Sensitivity with respect to shape changes. In level set-based topology optimisation, level set functions are updated in each iteration based on the sensitivity analysis of the shape-dependent optimisation objective function J. (For brevity, we present the derivations only for the objective function; the extension to constrained problems follows analogously.) We assume that J is shape-differentiable, meaning that for any level set-based shape update Ω_ε of a shape Ω generated by the vector field Θ_ε, the limit

$$DJ_\Omega(\Theta_{\varepsilon=0}) := \lim_{\varepsilon \to 0} \frac{J(\Omega_\varepsilon) - J(\Omega)}{\varepsilon} \tag{2}$$

exists and is a linear function of Θ_ε (making it a so-called Fréchet derivative). The above limit is called the *shape derivative* of J with respect to the given shape update.

It can be shown using the techniques of shape differentiation (see [11] and numerous papers on level set-based topology optimisation e.g., [4,28]) that the shape derivative of many shape-differentiable shape functions has the form

$$D\mathcal{J}_\Omega(\Theta_{\varepsilon=0}) = \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta_{\varepsilon=0}, \tag{3}$$

where the scalar-valued function $d\mathcal{J}_\Omega : \partial\Omega \to \mathbb{R}$ is known as the *shape gradient* of $\mathcal{J}$ at the shape Ω. The form (3) is an instance of a general result known as the *Hadamard-Zolésio structure theorem*. Moreover, one can often derive a reasonably explicit formula for $d\mathcal{J}_\Omega$ depending on the specific details of the function $\mathcal{J}$. We will not present any derivations here since these can readily be found in the literature; see for example [3].

Level set-based shape updates that are descent directions for an objective function. Typically, in level set-based optimisation algorithms, the level set updates are required to cause the objective function $\mathcal{J}$ to *decrease* in every iteration. The above sensitivity analysis provides a condition on Θ_ε that ensures this for sufficiently small ε as the shape is updated under the deformation generated by Θ_ε. We can deduce this condition from the identity

$$\mathcal{J}(\Omega_\varepsilon) = \mathcal{J}(\Omega) + \varepsilon D\mathcal{J}_\Omega(\Theta_{\varepsilon=0}) + o(\varepsilon), \tag{4}$$

which holds by definition of shape-differentiability. We see that $\mathcal{J}(\Omega_\varepsilon)$ decreases for sufficiently small ε when

$$D\mathcal{J}_\Omega(\Theta_{\varepsilon=0}) = \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta_{\varepsilon=0} < 0 \tag{5}$$

is satisfied. A shape update satisfying this condition is said to be a *descent direction* for $\mathcal{J}$ at Ω. In particular, the *steepest descent direction* is when the shape update satisfies $n_{\partial\Omega} \cdot \Theta_{\varepsilon=0} = -d\mathcal{J}_\Omega$.

Velocity extension. Being a descent direction is clearly an underdetermined condition because (5) constrains *only* the values of the normal component of $\Theta_\varepsilon(x)$ for $x \in \partial\Omega$ at time $\varepsilon = 0$. The final step in the update of the level set function via the transport Eq. (1) is thus to *extend* the definition of $\Theta_\varepsilon(x)$ to all remaining x (at least in a narrowband of $\partial\Omega$ and for $\varepsilon \neq 0$).

There are two common ways to construct such extensions, the classical *normal extension procedure* where Θ_ε is defined to be constant along normal lines emanating from $\partial\Omega$ in the narrowband, and the *HSE procedure*, where Θ_ε is obtained by solving a partial differential equation (PDE) in the narrowband. The latter has been found to be more desirable since it creates a smoother extension velocity and leads to fewer oscillations in the overall convergence of the topology optimisation procedure. See [10] for a discussion of these and other extension approaches.

2.3. The Hilbert space extension method

We will now describe the HSE method in more detail, since the approach we propose in this paper for constructing the extension velocity in such a way that it preserves the nature of the interfaces is based on it.

The linear algebra of gradients in a Hilbert space. A *Hilbert space* is a complete vector space that possesses a symmetric, strictly positive definite bilinear form, a.k.a., an inner product. Let $\mathcal{H}$ be a Hilbert space with an inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathcal{H}}$. Consider the Fréchet derivative of a function $T : \mathcal{H} \to \mathbb{R}$, namely the linear functional DT_u defined by

$$DT_u(v) := \frac{d}{d\varepsilon} T(u + \varepsilon v)\bigg|_{\varepsilon=0} \qquad \forall\, v \in \mathcal{H}$$

that gives the rate of change of T in the direction v at the point $u \in \mathcal{H}$. The so-called *Hilbert space gradient* of T at $u \in \mathcal{H}$, denoted $\mathrm{grad}(T)_u$, satisfies

$$DT_u(v) = \langle\!\langle \mathrm{grad}(T)_u, v \rangle\!\rangle_{\mathcal{H}} \qquad \forall\, v \in \mathcal{H}.$$

Using the properties of the Hilbert space inner product and the definition of the Fréchet derivative, it is straightforward to show that $-\mathrm{grad}(T)_u$ is a descent direction for T at $u \in \mathcal{H}$ (see [3]).

Application to the shape gradient in level set-based topology optimisation. We now consider a Hilbert space of infinitesimal generators of deformations of a domain $\mathcal{B} \subseteq \mathcal{D}$ contained in the universal design domain $\mathcal{D}$. To be precise, $\mathcal{B}$ is a narrowband of $\partial\Omega$ from which we have excised smaller narrowbands of any parts of Ω where the extension velocity is required to vanish (e.g., non-design regions in the interior of Ω or on the boundary $\partial\Omega$). Let Γ_0 be the union of all parts of $\partial\mathcal{B}$ in contact with these non-design regions. We choose the Hilbert space $H^1_0(\mathcal{B}, \mathbb{R}^3)$ of square-integrable vector fields on $\mathcal{B}$ with square-integrable first weak derivatives and satisfying homogeneous Dirichlet boundary conditions on Γ_0. We equip this space with the inner product

$$\langle\!\langle \Theta_1, \Theta_2 \rangle\!\rangle_\gamma := \int_{\mathcal{B}} \Theta_1 \cdot \Theta_2 + \gamma^2 \int_{\mathcal{B}} D\Theta_1 : D\Theta_2 \qquad \forall \Theta_1, \Theta_2 \in H^1_0(\mathcal{B}, \mathbb{R}^3),$$

where $\gamma > 0$ is a scalar parameter. The space $H^1_0(\mathcal{B}, \mathbb{R}^3)$ together with the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle_\gamma$ is an example of a so-called *Sobolev space* of vector-valued functions. Next, we consider the shape derivative of the objective function $\mathcal{J}$ as a linear functional on this space, namely

$$D\mathcal{J}_\Omega : H^1_0(\mathcal{B}, \mathbb{R}^3) \to \mathbb{R}$$
$$D\mathcal{J}_\Omega(\Theta) := \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta,$$

where $d\mathcal{J}_\Omega$ is the shape gradient as defined in Section 2.2. We note that $D\mathcal{J}_\Omega$ is indeed a bounded linear functional on $H^1_0(\mathcal{B}, \mathbb{R}^3)$ under suitable regularity assumptions thanks to the *trace theorems* of mathematical analysis [14]. Consequently, $D\mathcal{J}_\Omega$ has a Hilbert space gradient with respect to the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle_\gamma$, which satisfies

$$\langle\!\langle \mathrm{grad}(\mathcal{J})_\Omega, \Theta \rangle\!\rangle_\gamma = \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta \qquad \forall\, \Theta \in H^1_0(\mathcal{B}, \mathbb{R}^3). \tag{6}$$

As discussed above, the vector field $-\mathrm{grad}(\mathcal{J})_\Omega$ is a descent direction for $\mathcal{J}$ at Ω. That is, if Ω_ε denotes the updated shape with respect to the deformation generated by this vector field, then we substitute $\Theta_{\varepsilon=0} := -\mathrm{grad}(\mathcal{J})_\Omega$ into (4) and apply (6) to obtain

$$\mathcal{J}(\Omega_\varepsilon) = \mathcal{J}(\Omega) - \varepsilon \langle\!\langle \mathrm{grad}(\mathcal{J})_\Omega, \mathrm{grad}(\mathcal{J})_\Omega \rangle\!\rangle_\gamma + o(\varepsilon) < \mathcal{J}(\Omega)$$

when $\varepsilon > 0$ is sufficiently small. In the context of topology optimisation, we will call the vector field $\mathrm{grad}(\mathcal{J})_\Omega$ the *HSE velocity* associated with the shape derivative $D\mathcal{J}_\Omega$. Moreover, we will see below why the HSE velocity enjoys a smoothing property controlled by γ. As mentioned, this smoothing property has been observed to improve the behaviour of the shape updates, leading to better convergence in level set-based topology optimisation.

NOTE: There are many ways to apply the notion of Hilbert space gradient to the shape derivative of a shape-differentiable objective function. Some interesting possibilities are described in [3], including an interpretation of the Sequential Linear Programming approach to level set-based topology optimisation [28].

Equations satisfied by the HSE velocity. The defining Eq. (6) of the HSE velocity amounts to the weak form of a PDE. It is instructive to see the PDE itself, which we obtain in the usual way by integrating the

derivative terms by parts. We see that $\Theta := \text{grad}(\mathcal{J})_\Omega$ satisfies

$$\begin{aligned} \Theta - \gamma^2 \Delta \Theta &= G \qquad \text{in } \mathcal{B}, \\ \frac{\partial \Theta}{\partial n} &= 0 \qquad \text{on } \partial\mathcal{B} \setminus \Gamma_0, \\ \Theta &= 0 \qquad \text{on } \Gamma_0, \end{aligned} \tag{7}$$

where Δ is the scalar Laplace operator and $\frac{\partial}{\partial n}$ is the outward normal derivative, both acting component-wise. The right-hand side G should be understood as a *distribution* supported on $\partial\Omega$ which is defined by integration against test functions via

$$\int_{\mathcal{B}} G \cdot \Theta := \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta \qquad \forall \Theta \in H_0^1(\mathcal{B}, \mathbb{R}^3).$$

From the strong form (7), we see that the components of the HSE velocity satisfy an elliptic PDE whose eigenvalues are strictly positive thanks to the presence of the negative Laplace operator. Therefore, we can interpret the solution of (7) as a diffusive *smoothing* of G where γ can be characterised as a length-scale over which the smoothing takes place.

3. Geometrically constrained level set-based shape updates

We now turn to the central construction of this paper, namely a shape update for a component in a larger assembly that preserves the desired geometric features of the interface regions where the component interacts with its neighbouring components. In the context of level set-based topology optimisation, this means constructing the infinitesimal generator Θ_ε of a level set update, via the transport Eq. (1), that achieves this behaviour in the updated level set function F_ε. We propose the following two-step construction:

1. Enforce infinitesimal motion constraints in the interface regions during the velocity extension procedure that determines $\Theta_{\varepsilon=0}$. The resulting vector field must be a descent direction for the optimisation objective function.
2. Define Θ_ε for $\varepsilon \neq 0$ in such a way that the transport equation induces the desired macroscopic motion of the level set function.

We now describe these two steps in more detail, beginning with precise definitions of the motions we wish to allow in the interface regions.

3.1. Geometry of the interfaces

The shapes we consider in this paper are meant to represent the geometry of an individual component in an assembly consisting of multiple interacting components. A shape Ω of this kind interacts with the rest of the assembly through a collection of *interfaces* where Ω comes into contact with its neighbours. Thus, we assume $\partial\Omega$ contains a collection of disjoint subsets $\Gamma_I^{(1)}, \ldots, \Gamma_I^{(N)}$ where these interactions occur. As we know, the geometry of these subsets is special — e.g., flat if the interface supports a weld, cylindrical if the interface supports a revolute joint. We endow the n^{th} interface with a spatial location parameter $p_I^{(n)} \in \mathbb{R}^3$ and an orientation parameter $Q_I^{(n)} \in \mathbb{R}^{3\times3}$ given as an orthonormal matrix; these are sufficient to locate the interfaces in space. Fig. 2 illustrates these parameters for a domain with two interfaces.

NOTE: Recall that $\partial\Omega$ may also contain additional subsets where external surface traction forces per unit area (i.e., Neumann boundary conditions) or fixities (i.e., Dirichlet boundary conditions) are imposed. We denote these subsets by Γ_N and Γ_D, respectively. If we wish to include these in the design domain, we count them among the interfaces.

3.2. Preserving geometry at the interfaces

Thanks to their critical role in the assembly, component interfaces are only allowed to move during topology optimisation in such a way as to preserve their special nature — e.g., flat interfaces remain flat, cylindrical interfaces remain cylindrical. In this paper, we satisfy this requirement by ensuring that the interfaces are only allowed to move by rigid motions (translations and rotations) and by orthotropic scalings aligned with the orientation of the interface (e.g., cylindrical aperture supporting a revolute joint may increase in radius or length). Motions of this kind will affect the location and orientation parameters of the interfaces as well as their sizes relative to the axes of orientation, but will not change the nature of the interfaces. In what follows, we show how to define and impose these motion constraints.

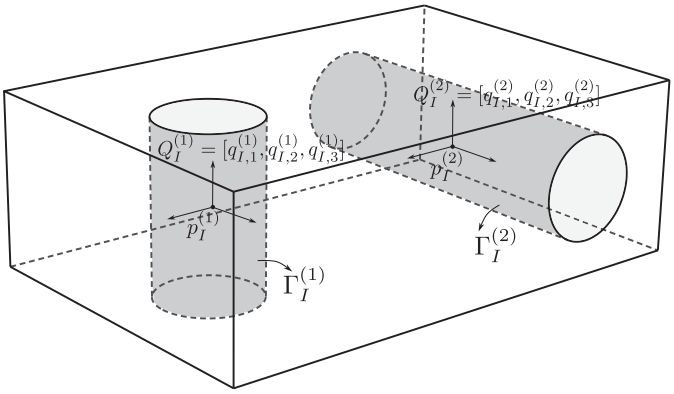


Fig. 2. Notation for interfaces.

Allowed motions. We state the following definitions of the allowable motions and their infinitesimal generators.

Translations. The one-parameter family of translations by $t \in \mathbb{R}^3$ is given by

$$\phi_\varepsilon^{trans}(x) := x + \varepsilon t.$$

The infinitesimal generator of this motion is the affine vector field

$$\Theta^{trans}(x) := t.$$

Without loss of generality, we can write $t = Qt_0$ for $t_0 \in \mathbb{R}^3$ when an orientation is prescribed via an orthogonal matrix $Q \in \mathbb{R}^{3\times3}$.

Rotations. The one-parameter family of rotations about a fixed point $p \in \mathbb{R}^3$ and an axis $h \in \mathbb{R}^3$ through p is given by

$$\phi_\varepsilon^{rot}(x) := \exp(\varepsilon \mathcal{A}(h))(x - p) + p,$$

where $\mathcal{A}(h)$ is the antisymmetric matrix satisfying $\mathcal{A}(h)x = h \times x$. The matrix exponential can be computed using the Rodrigues formula. Note that h is un-normalised, and thus $\|h\|$ gives the angular velocity of the rotation about the unit axis $h/\|h\|$. The infinitesimal generator of this motion is the affine vector field

$$\Theta^{rot}(x) := \mathcal{A}(h)(x - p).$$

Without loss of generality, we can write $h = Qh_0$ for $h_0 \in \mathbb{R}^3$ when an orientation is prescribed via an orthogonal matrix $Q \in \mathbb{R}^{3\times3}$.

Orthotropic scalings. The one-parameter family of orthotropic scalings about a fixed point p and by scaling factors $d := (d_1, d_2, d_3)$ oriented along the columns of an orthogonal matrix $Q \in \mathbb{R}^{3\times3}$ is given by

$$\phi_\varepsilon^{scal}(x) := Q \exp(\varepsilon D) Q^\top (x - p) + p,$$

where $D = diag(d)$ and the matrix exponential is just $\exp(\varepsilon D) = diag(e^{\varepsilon d_1}, e^{\varepsilon d_2}, e^{\varepsilon d_3})$. The infinitesimal generator of this motion is the

affine vector field

$$\Theta^{scal}(x) := QDQ^{\top}(x-p).$$

Compound motions. It is necessary to make a concrete choice for the order of operations of a combination of the above motions since they do not, in general, commute. We choose

$$\begin{aligned}\phi_{\varepsilon}^{allowed}(x) &:= \phi_{\varepsilon}^{trans}\circ\phi_{\varepsilon}^{rot}\circ\phi_{\varepsilon}^{scal}(x) \\ &= \exp(\varepsilon\mathcal{A}(h))Q\exp(\varepsilon D)Q^{\top}(x-p)+p+\varepsilon t\end{aligned} \tag{8}$$

as the allowed compound motion of each interface. The infinitesimal generator of this motion is the affine vector field

$$\begin{aligned}\Theta_{\varepsilon}^{allowed}(x) &:= t+\mathcal{A}(h)(x-p-\varepsilon t) \\ &\quad+\exp(\varepsilon\mathcal{A}(h))\,QDQ^{\top}\exp(-\varepsilon\mathcal{A}(h))(x-p-\varepsilon t).\end{aligned} \tag{9}$$

Note that unlike the infinitesimal generators of the individual translations, rotations, and orthotropic scalings, this vector field is explicitly time-dependent.

A basis for the allowed infinitesimal generators of an interface. We see from (9) that the infinitesimal generator of the allowed compound motion at $\varepsilon = 0$ satisfies

$$\Theta_{\varepsilon=0}^{allowed}(x) := t+\mathcal{A}(h)(x-p)+QDQ^{\top}(x-p), \tag{10}$$

which is simply a linear combination of an infinitesimal translation field, an infinitesimal rotation field, and an infinitesimal orthotropic scaling field. Therefore, the allowed infinitesimal motions belong to a finite-dimensional space spanned by vector fields of these kinds. We now construct a basis for this space near a given interface.

Let Γ_I be one of the interfaces, with the spatial location parameter p_I and orientation parameter $Q_I := [q_{I,1}, q_{I,2}, q_{I,3}]$. Define the linearly independent vector fields

$$\begin{aligned}&V_1^{trans}(x) := q_{I,1}, \quad V_1^{rot}(x) := \mathcal{A}(q_{I,1})(x-p_I), \quad V_1^{scal}(x) := q_{I,1}[q_{I,1}]^{\top}(x-p_I),\\ &V_2^{trans}(x) := q_{I,2}, \quad V_2^{rot}(x) := \mathcal{A}(q_{I,2})(x-p_I), \quad V_2^{scal}(x) := q_{I,2}[q_{I,2}]^{\top}(x-p_I),\\ &V_3^{trans}(x) := q_{I,3}, \quad V_3^{rot}(x) := \mathcal{A}(q_{I,2})(x-p_I), \quad V_3^{scal}(x) := q_{I,3}[q_{I,3}]^{\top}(x-p_I).\end{aligned}$$

These vector fields form a basis for the nine-dimensional vector space spanned by the infinitesimal generators of all translations, rotations, and orthotropic scalings aligned with the orientation and location of the interface.

The desired vector space of allowed motions is a *subspace* of this vector space whose precise definition depends on the nature of the interface, since we may not always want to allow all nine dimensions of design freedom. For example, if the interface supports a revolute joint, then Γ_I is a cylinder oriented along, say, the axis $q_{I,1}$. The allowed motions must preserve the fact that Γ_I is a cylinder; thus only orthotropic scalings in the $q_{I,1}$ direction and uniform orthotropic scalings in the $\{q_{I,2}, q_{I,3}\}$ plane are allowed, while rotations about the $q_{I,1}$ axis are irrelevant. Therefore, the subspace of allowed motions in this case is the seven-dimensional vector space spanned by the velocity fields $V_1^{trans}, V_2^{trans}, V_3^{trans}, V_2^{rot}, V_3^{rot}, V_1^{scal}$, and $V_2^{scal}+V_3^{scal}$.

Allowed motion constraints on the extension velocity. Recall that in each iteration of topology optimisation, the motion of the shape Ω is obtained by solving the transport Eq. (1) for the level set update with an extension velocity Θ_ε derived from a sensitivity calculation. We will thus achieve the allowable motions of each interface of Ω by imposing constraints on Θ_ε in the vicinity of each interface.

Let $\mathcal{B}$ be the narrowband of $\partial\Omega$ where we intend to construct the extension velocity, and let $\mathcal{N}_I^{(n)} \subseteq \mathcal{B}$ denote a smaller narrowband of each $\Gamma_I^{(n)}$ that is small enough to be contained in $\mathcal{B}$. We can now formulate the allowed motion constraints as follows.

- First, we require that the restriction of $\Theta_{\varepsilon=0}$ to this narrowband belongs to the vector space spanned by the allowed infinitesimal motions of $\Gamma_I^{(n)}$. Let $V_1^{(n)}, \dots, V_{d_n}^{(n)}$ for some $d_n \le 9$ be a basis for the allowed infinitesimal motions near $\Gamma_I^{(n)}$. Thus, there exist scalar coefficients $z_1^{(n)}, \dots, z_{d_n}^{(n)} \in \mathbb{R}$ so that

$$\Theta_{\varepsilon=0}(x) = \sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)}(x) \qquad \forall\, x \in \mathcal{N}_I^{(n)}. \tag{11}$$

  Each such vector can then be expressed as a linear combination of the infinitesimal rotation, translation, and scaling vector fields introduced above.
- Second, we require that $\Theta_{\varepsilon\neq 0}$ has the form (9), where t, h, and d are the appropriate linear combinations of the z-coefficients determined from (11).

These constraints now ensure that the deformation generated by Θ_ε performs the allowed motion near $\Gamma_I^{(n)}$ for sufficiently small ε — namely until $\phi_\varepsilon(x)$ leaves $\mathcal{N}_I^{(n)}$ for some $x \in \Gamma_I^{(n)}$.

3.3. The constrained Hilbert space extension method

We now introduce the *constrained Hilbert space extension method* as the first step in the construction of the extension velocity that we propose to use in each iteration of level set-based topology optimisation. The purpose of this method is to find an update velocity that satisfies the constraints (11) while remaining a descent direction for the optimisation objective function and enjoying the smoothing properties of the conventional HSE method.

The HSE procedure is variational. The starting point for our modification of the HSE procedure is the following important observation: the conventional HSE velocity associated with the shape gradient $d\mathcal{J}_\Omega$ of a shape-differentiable objective function $\mathcal{J}$ can be obtained by solving a variational problem. That is, Equation (6) defining the HSE velocity is the first-order optimality condition for the variational problem

$$\begin{aligned}\text{minimise} \quad & \frac{1}{2}\langle\!\langle\Theta,\Theta\rangle\!\rangle_\gamma - \int_{\partial\Omega} d\mathcal{J}_\Omega\, n_{\partial\Omega}\cdot\Theta \\ \text{over} \quad & \Theta \in H_0^1(\mathcal{B},\mathbb{R}^3).\end{aligned}$$

Since the above minimisation problem is convex quadratic, the solution of the first-order optimality condition is the unique global minimiser.

Adding allowed motion constraints. Our proposed method is simply to include the constraints (11) on the extension velocity into the above minimisation problem. That is, we define the desired extension velocity Θ_{CHSE} to be the solution of

$$\text{minimise} \quad \frac{1}{2}\langle\!\langle\Theta,\Theta\rangle\!\rangle_\gamma - \int_{\partial\Omega} d\mathcal{J}_\Omega\, n_{\partial\Omega}\cdot\Theta \tag{12a}$$

$$\text{subject to} \quad \Theta(x) = \sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)}(x) \qquad \forall\, x \in \mathcal{N}_I^{(n)} \text{and } \forall\, n = 1,\dots N \tag{12b}$$

$$\text{over} \quad \Theta \in H_0^1(\mathcal{B},\mathbb{R}^3) \quad \text{and} \quad z_1^{(1)},\dots,z_{d_N}^{(N)} \in \mathbb{R}.$$

Since the included constraints carve out a linear subspace of $H_0^1(\mathcal{B},\mathbb{R}^3)$, the constrained minimisation problem is still convex and has a unique global minimiser.

Equations satisfied by the C-HSE velocity. We first recast the variational problem (12) in a more convenient form. It is clear that (12) is equivalent to minimising the objective function (12a) over vector fields defined on

the smaller domain

$$\mathcal{B}' := \mathcal{B} \setminus \bigcup_{n=1}^{N} \mathcal{N}_I^{(n)}$$

with the constraints (12b) given as boundary conditions on each $\partial\mathcal{N}_I^{(n)}$. The minimiser in the full domain $\mathcal{B}$ is then just obtained by extending the solution to the interior of each $\mathcal{N}_I^{(n)}$ via (12b) with the minimising z-coefficients.

We would typically include *essential* boundary conditions of this form (12b) in the definition of the space of functions over which the variational problem is posed. However, since the unknown z-coefficients appear in these boundary conditions, it turns out to be more convenient to include them in the objective function instead via a Nitsche term [19]. Thus, we arrive at the equivalent variational problem for the C-HSE velocity

$$\begin{aligned} \text{minimise} \quad & \tfrac{1}{2}\mathcal{Q}\big(\Theta, z_1^{(1)}, \dots, z_{d_N}^{(N)}\big) - \mathcal{L}\big(\Theta, z_1^{(1)}, \dots, z_{d_N}^{(N)}\big) \\ & - \textstyle\sum_{n=1}^{N} \int_{\partial\mathcal{N}_I^{(n)}} \frac{\partial\Theta}{\partial n} \cdot \Big(\Theta - \sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)}\Big) \\ \text{over} \quad & \Theta \in H_0^1(\mathcal{B}', \mathbb{R}^3) \quad \text{and} \quad z_1^{(1)}, \dots, z_{d_N}^{(N)} \in \mathbb{R}, \end{aligned} \tag{13}$$

where

$$\begin{aligned} \mathcal{Q}\big(\Theta, z_1^{(1)}, \dots, z_{d_N}^{(N)}\big) &:= \int_{\mathcal{B}'} \Theta \cdot \Theta + \gamma^2 \int_{\mathcal{B}'} D\Theta : D\Theta \\ &\quad + \sum_{n=1}^{N} \sum_{s,t=1}^{d_n} z_s^{(n)} z_t^{(n)} \int_{\mathcal{N}_I^{(n)}} \Big(V_s^{(n)} \cdot V_t^{(n)} + \gamma^2 DV_s^{(n)} : DV_t^{(n)} \Big), \\ \mathcal{L}\big(\Theta, z_1^{(1)}, \dots, z_{d_N}^{(N)}\big) &:= \int_{\partial\Omega \cap \mathcal{B}'} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta + \sum_{n=1}^{N} \sum_{s=1}^{d_n} z_s^{(n)} \int_{\partial\Omega \cap \mathcal{N}_I^{(n)}} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot V_s^{(n)} \end{aligned}$$

are, respectively, positive-definite quadratic and linear functionals acting on both functions $\Theta \in H_0^1(\mathcal{B}', \mathbb{R}^3)$ as well as the z-coefficients.

We can now derive the weak and strong forms of the first-order optimality conditions satisfied by minimising $\Theta \in H_0^1(\mathcal{B}', \mathbb{R}^3)$ as well as the z-coefficients in the usual way. For brevity, we present the weak form in Appendix A. The strong form consists of the following algebraic and partial differential equations:

$$\begin{aligned} \Theta - \gamma^2 \Delta\Theta &= G && \text{in } \mathcal{B}', \\ \Theta &= 0 && \text{on } \Gamma_0, \\ \Theta &= \sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)} && \text{on } \partial\mathcal{N}_I^{(n)} \text{for } n = 1, \dots, N, \\ \frac{\partial\Theta}{\partial n} &= 0 && \text{on } \partial\mathcal{B} \setminus \Big(\Gamma_0 \cup \partial\mathcal{N}_I^{(1)} \cup \cdots \cup \partial\mathcal{N}_I^{(N)}\Big), \end{aligned} \tag{14a}$$

together with

$$\sum_{t=1}^{d_n} K_{st}^{(n)} z_t^{(n)} + \int_{\partial\mathcal{N}_I^{(n)}} \frac{\partial\Theta}{\partial n} \cdot V_s^{(n)} = G_s^{(n)} \qquad \text{for } s = 1, \dots, d_n \text{and } n = 1, \dots, N. \tag{14b}$$

In the above equations,

$$\begin{aligned} K_{st}^{(n)} &:= \int_{\mathcal{N}_I^{(n)}} \Big(V_s^{(n)} \cdot V_t^{(n)} + \gamma^2 DV_s^{(n)} : DV_t^{(n)} \Big), \\ G_s^{(n)} &:= \int_{\partial\Omega \cap \mathcal{N}_I^{(n)}} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot V_s^{(n)}, \end{aligned}$$

and G is the distribution defined by its action on test functions via

$$\int_{\mathcal{B}'} G \cdot \delta\Theta := \int_{\partial\Omega \cap \mathcal{B}'} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \delta\Theta \qquad \forall\, \delta\Theta \in H_0^1(\mathcal{B}', \mathbb{R}^3).$$

If we now compare these equations with those satisfied by the unconstrained HSE velocity (7) we see that the C-HSE velocity satisfies the same elliptic partial differential Eq. (14a) but with new boundary conditions depending on the z-coefficients on each $\partial\mathcal{N}_I^{(n)}$. A unique solution to this equation exists for any choice of z-coefficients and exhibits the expected smoothing properties. The z-coefficients can then be determined uniquely via the integral conditions (14b). For details on how we discretise and solve (14) using the finite element method, see Appendix B.

The C-HSE velocity yields a descent direction. Let Θ_{CHSE} and $z_1^{(1)}, \dots, z_{d_N}^{(N)}$ be the solution of the C-HSE equations. Substitute these and $\delta\Theta := \Theta_{CHSE}$ and $\delta z_s^{(n)} := z_s^{(n)}$ into the weak Eq. (A.19). Thanks to the boundary conditions satisfied by Θ_{CHSE} on each $\partial\mathcal{N}_I^{(n)}$, the constraint terms vanish, and the remaining terms collapse by definition. Thus, we obtain

$$0 = \langle\!\langle \Theta_{CHSE}, \Theta_{CHSE} \rangle\!\rangle_\gamma - \int_{\partial\Omega} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \Theta_{CHSE}, \tag{15}$$

where we view Θ_{CHSE} as a function defined on $\mathcal{B}$, but equal to $\sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)}$ on each $\mathcal{N}_I^{(n)}$. Equation (15) implies that $-\Theta_{CHSE}$ is a descent direction for $\mathcal{J}$ at Ω.

Conclusion. The first step in the construction of the extension velocity used in each iteration of level set-based topology optimisation is to let $\Theta_{\varepsilon=0} := -\Theta_{CHSE}$, i.e., the negative C-HSE velocity defined in the previous paragraph. As a result, we obtain both a descent direction for the optimisation objective function and a velocity that satisfies the infinitesimal motion constraints (11) in the neighbourhood of each interface by construction.

3.4. Time-dependence of the extension velocity

We now turn to the second step of the construction of the extension velocity used in each iteration of level set-based topology optimisation. This is to define Θ_ε for $\varepsilon \neq 0$ in such a way that we obtain the allowed macroscopic motions of the interfaces, i.e., the combination of translation, rotation, and scaling of the form (8), at least for sufficiently small ε.

Recall that the desired motion will occur if the infinitesimal generator has the form (9) near each interface. We achieve this as follows. For each interface, let $t_I^{(n)}$, $h_I^{(n)}$, and $d_I^{(n)}$ be the unique translation, rotation, and scaling parameters determined from the coefficients $z_1^{(n)}, \dots, z_{d_n}^{(n)}$ to cause (11) to match the desired form inside $\mathcal{N}_I^{(n)}$ at $\varepsilon = 0$. Let $\Theta_\varepsilon^{allowed,(n)}$ denote the matching time-dependent vector field. Let $\chi_I^{(n)} : \mathcal{B} \to \mathbb{R}$ be a smooth, positive *cut-off function* that equals one on $\mathcal{N}_I^{(n)}$ and vanishes outside a slightly larger narrowband of $\Gamma_I^{(n)}$. Then, we define

$$\Theta_\varepsilon := \Theta_{CHSE} + \sum_{n=1}^{N} \chi_I^{(n)} \big(\Theta_\varepsilon^{allowed,(n)} - \Theta_{CHSE} \big), \tag{16}$$

where Θ_{CHSE} is the solution of (12). The above definition completes our construction of the interface-preserving extension velocity.

Thanks to our construction, the extension velocity given by (16) is explicitly time-dependent only within the support of each $\chi_I^{(n)}$, and coincides with the infinitesimal generator of the desired compound motion inside a slightly smaller subset of $\mathcal{N}_I^{(n)}$. Furthermore, it is continuous at $\varepsilon = 0$ because $\Theta_{\varepsilon=0}^{(n)} = \Theta_{CHSE}$ inside the interface region $\mathcal{N}_I^{(n)}$, which is the support of $\chi_I^{(n)}$. Therefore, the solution of the transport equation with time-dependent infinitesimal generator Θ_ε will produce a continuous evolution of the level set function everywhere and generate the desired motion near each interface, at least for sufficiently small ε.

NOTE: It is not always necessary to carry out this step. This is because for certain combinations of vector subspaces of allowed motions and interface geometries, the time-independent vector field (11) integrates into a one-parameter family of motions that preserves the interface geometry, although it does not explicitly have the sequential form (8).

Therefore, in such a case, no special definition for $\epsilon > 0$ needs to be made. In particular, if the allowed motions consist of translations and rotations only, or translations and scalings only, then it is possible to integrate the time-independent vector field to create the desired motion. When scalings and rotations are combined, however, it is necessary to be more careful since the non-commutativity of these operations can lead to undesired shearing when the time-independent vector field is used. We will not analyse this phenomenon further here, since that would take us too far afield.

3.5. A caveat on topology changes

It is important to note that the use of the linear transport equation to update the level set function with respect to a background time-dependent velocity field precludes topology changes because the evolving level sets are all diffeomorphic to each other under the deformation generated by this velocity field. One can view this as a limitation of our method, because the ability to induce topology changes is seen as critical for obtaining "true" optimised geometry that is useful from a designer's perspective, rather than local optima of the objective function.

In this paper, we have decided not to address this limitation because our goal is simply to demonstrate that a level set update with a well-chosen velocity field preserves the nature of the interfaces while allowing their spatial location and scale to be optimised. Nevertheless, we suggest here a more elaborate procedure for integrating the C-HSE velocity that does not preclude topology changes outside the interface regions.

Topology changes in standard topology optimisation arise because the level set is updated with respect to an *adaptive* velocity field — one that remains normal to the evolving level sets while having a prescribed speed derived from sensitivity analysis. With this in mind, we suggest a small modification of (16) to create an adaptive velocity outside the interface regions while maintaining the required velocity near the interfaces. Let $n_\epsilon := \nabla F_\epsilon / \|\nabla F_\epsilon\|$ be the unit normal vector field of the level sets, and let $\Theta^\perp_{CHSE} := \langle \Theta_{CHSE}, n_\epsilon \rangle$ be the normal component of Θ_{CHSE}. Then, we propose

$$\Theta_\epsilon^{modified} := \Theta^\perp_{CHSE}\, n_\epsilon + \sum_{n=1}^{N} \chi_I^{(n)} \big(\Theta_\epsilon^{allowed,(n)} - \Theta^\perp_{CHSE}\, n_\epsilon \big).$$

If we now substitute the modified velocity into the transport equation for the evolving level set field F_ϵ, we obtain

$$\frac{\partial F_\epsilon}{\partial \epsilon} + \Theta^\perp_{CHSE} \|\nabla F_\epsilon\| + \sum_{n=1}^{N} \chi_I^{(n)} \big\langle \big(\Theta_\epsilon^{allowed,(n)} - \Theta^\perp_{CHSE}\, n_\epsilon \big), \nabla F_\epsilon \big\rangle = 0$$

$$F_0 = F.$$

We see that outside of the interface regions, all the cut-off functions vanish, and we recover the level set update of standard topology optimisation with prescribed speed equal to $\Theta^\perp_{CHSE}$. This transitions smoothly until we are inside any of the interface regions, where one cut-off function is identically equal to one. There, we have the level set update via the transport equation with an interface-preserving velocity field. We will test this modified update procedure in future work.

4. Application to topology optimisation

4.1. A representative topology optimisation problem

In this section, we apply the shape update procedure we have developed to a collection of archetypal topology optimisation problems. Although our shape update is eventually meant to be used to optimise one or more components belonging to an assembly of multiple components interacting with each other through interfaces that support mechanical joints, we restrict ourselves for the moment to a single-component context, where the boundary of the shape to be designed has one or more interfaces of a specified nature. This allows us to demonstrate the utility of our update procedure while avoiding the additional complications of the multiple-component context, e.g., full-assembly simulation and the associated sensitivity analysis, and multiple co-evolving level set functions. Nevertheless, we attempt to choose topology optimisation scenarios that are as representative as possible of the multiple-component context.

Problem statement. We now state a common formulation for the collection of topology optimisation problems that we solve below. We will find an optimal shape from an admissible class of shapes, where a shape Ω is deemed admissible if $\partial\Omega$ contains interfaces $\Gamma_I^{(1)}, \dots, \Gamma_I^{(N)}$. The geometry of these interfaces is specified in advance, parameterized by their location, orientation in space, and scale. Each interface is either designable or non-designable, and only the designable ones will have their parameter values determined through optimisation.

The optimal shape and optimal interface parameter values are determined by minimising the *elastic compliance* with respect to a single set of external loads and fixities. We also subject the optimal shape to an equality constraint on its volume. We solve this equality-constrained topology optimisation problem using the augmented Lagrangian algorithm [20] (a common approach in level set-based topology optimisation), where the shape is updated in each inner iteration of this algorithm with our shape update procedure applied to the augmented Lagrangian shape function.

The elastic compliance is defined as follows. Let Γ_{load} denote the union of all the interfaces of Ω where loads are applied. We denote these loads collectively by the load distribution per unit area $g : \Gamma_{load} \to \mathbb{R}^3$. Let Γ_{fix} denote the union of all the interfaces of Ω where fixities are applied. Let $\Gamma_{free} := \partial\Omega \setminus (\Gamma_{load} \cup \Gamma_{fix})$ denote the remainder of the boundary. Additionally, we suppose that a body force distribution per unit volume $f : \Omega \to \mathbb{R}^3$ acts on Ω. The elastic compliance of a shape Ω with respect to this loading condition is given by

$$\mathcal{J}_{comp}(\Omega) := \int_\Omega \sigma(u_\Omega) : e(u_\Omega),$$

where u_Ω is the elastic response of Ω under these loads, $e(u_\Omega)$ is the associated strain tensor, and $\sigma(u_\Omega)$ is the associated stress tensor. The vector field u_Ω satisfies the linear elastic equation

$$\begin{aligned} -\mathrm{div}(\sigma(u_\Omega)) &= f && \text{in } \Omega, \\ n \cdot \sigma(u_\Omega) &= g && \text{on } \Gamma_{load}, \\ n \cdot \sigma(u_\Omega) &= 0 && \text{on } \Gamma_{free}, \\ u_\Omega &= 0 && \text{on } \Gamma_{fix}, \end{aligned} \tag{17}$$

where n is the outward-pointing unit normal vector field of $\partial\Omega$.

4.2. Sensitivity analysis

As mentioned in Section 2.2, the sensitivity of the elastic compliance with respect to a deformation Ω_ϵ generated by a vector field Θ_ϵ has the form

$$D\mathcal{J}_{comp,\Omega}(\Theta_{\epsilon=0}) := \frac{d}{d\epsilon} \mathcal{J}_{comp}(\Omega_\epsilon)\Big|_{\epsilon=0} = \int_{\partial\Omega} d\mathcal{J}_{comp,\Omega}\, n_{\partial\Omega} \cdot \Theta_{\epsilon=0}, \tag{18}$$

where $d\mathcal{J}_{comp,\Omega}$ is the shape gradient of the elastic compliance with respect to boundary variations. We must use (18) in order to compute the terms appearing in (B.21) for the right-hand side of the C-HSE Eqs. (B.22).

The formula for the shape gradient away from the interfaces supporting loads and fixities is well-known and can readily be found in the literature (e.g., [4]), namely

$$d\mathcal{J}_{comp,\Omega} = -\sigma(u_\Omega) : e(u_\Omega) + 2\, f \cdot u_\Omega \qquad \text{on } \Gamma_{free}.$$

We use this formula to compute the $G_\alpha^{(0)}$ source terms of (B.21). On the other hand, the formula for the shape gradient on the interfaces supporting loads and fixities when these are allowed to deform under the

action of Θ_ε is more complicated. First, one must be precise about what a load applied to a deforming interface means and include this in the calculation of the shape derivative. Second, one must take into account the singularities that appear at the curve of overlap between the free boundary and the interfaces where the fixities are applied [9]. We have decided to side-step these difficulties by computing the $G^{(n)}$ using finite-differencing. That is, for each deformation generated by the allowed infinitesimal motions for each interface, we approximate the derivative with respect to ε appearing in (18) by a difference quotient with suitably small ε. In our test cases, the number of interfaces is small, so this approach is not too onerous; but it is, of course, not optimal. For the moment, however, it serves the purpose of demonstrating our velocity-constrained shape updates. We will replace the finite-differencing with an approach suggested in [9] (i.e., the volumetric shape derivative formula) in future work.

4.3. Results

In this section, we demonstrate the C-HSE method on three representative problems in 3D: an L-bracket with an aperture, a cantilever, and the rocker component of the vehicle suspension system discussed in the introduction. These problems demonstrate the various geometric constraints that can be imposed on distinct regions of the domain, including loaded or fixed interfaces, with free-form optimisation in the rest of the domain.

4.3.1. Implementation details

Here, we note additional implementation details that, while they are not directly related to our novel contributions, have an important effect on the subsequent results presented here.

Meshing. Each time we need to evaluate the compliance of a new shape, we regenerate the mesh from the signed distance function representing the shape. This occurs for each new shape generation but also for each iteration of any line searches and for any finite difference computations. To generate a volume mesh for the component, we first generate a surface mesh from the signed distance function using OpenVDB [18]. We then use geogram [7] to remesh the surface mesh, including mesh refinement in the regions of loads or supports (this helps reduce the variance in the compliance due to meshing). Finally, we employ geogram again to generate the volume mesh from the refined surface mesh.

Timestep selection. Each time we generate a velocity field in order to update the shape, we need to decide on a timestep. There are a few mechanisms that inform this decision: first, at the start of each inner loop of the augmented Lagrangian (i.e., when the augmented Lagrangian parameters are updated), we perform a line search [20] with six iterations to determine an optimal timestep for that iteration. For subsequent iterations, we take the timestep value from the previous iteration, increase it by a small factor, and then apply it to the current iteration. Finally, in a procedure known as back-tracking, if the Lagrangian has increased after a shape update, we restore the component shape to the state from before the update, reduce the timestep, and update the shape again with the smaller timestep.

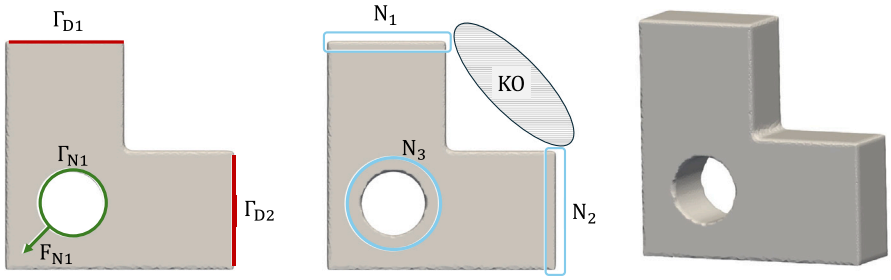
Smoothing parameter selection. The smoothing parameter γ is the foremost hyperparameter for the C-HSE method, controlling the amount of smoothing applied to the velocity field. We note that higher values of smoothing (to a certain extent) engender faster convergence, while lower values of γ lead to slightly lower objective function values at the end of optimisation. The smoothing parameter values for the following problems were selected heuristically.

Topology change. One interesting numerical artefact produced by our implementation is that it *does* produce topology changes in the zero-level set of the evolving level set function, contrary to what we expect from theory. This is evident in the results that we will present below. We speculate that the reason for this is the effective numerical dissipation present in the first-order finite-difference scheme of the transport equation, which allows the solution procedure to mimic the behaviour of the non-linear Hamilton-Jacobi equation of standard level set-based topology optimisation with respect to topology change. For the moment, we benefit from this artefact in practice, since it allows us to avoid the additional complexity described in Section 3.5. Nevertheless, we intend to investigate this artefact more thoroughly in future work.

4.3.2. L-bracket with an aperture

The initial geometry of the L-bracket is shown in Fig. 3. A load is applied to the face indicated by the green circle (in the direction indicated by the green arrow), and a fixed boundary condition is imposed on the two red surfaces. The blue circles represent constraint regions, each with different imposed geometric constraints. The constraint region labelled N_1 is allowed to scale and translate horizontally, while N_2 is allowed to scale and translate vertically. The region N_3 is allowed to translate in-plane. Finally, the region marked "KO" is defined as a keep-out (i.e., the component may not overlap with this region). We minimise the compliance of the shape under the defined load, with a volume constraint ensuring that the volume of our final component does not exceed 20 % of the initial volume. For this problem with a characteristic length of 140, the smoothing parameter γ^2 was set to 500 (the effect of this parameter depends on the size of the problem).

The evolution of the shape throughout the optimisation is shown in Fig. 4, and the final shape is shown in Fig. 5. Fig. 6 shows the convergence of the Lagrangian for this problem. This problem demonstrates some key features: first, our ability to very closely maintain the shape of the constrained regions (the loaded face Γ_{N_1} remained perfectly cylindrical); second, the simultaneous scaling and translation of constraint regions; third, our ability to optimise for the position and size of the fixities. We remark that despite the symmetry of the setup for this problem, the optimised shape (Fig. 5) is non-symmetric. While this was mildly unexpected, the non-symmetry seems to aid in minimising the compliance,



(a) Fixed interfaces (red), loaded interfaces (green) and applied loads.

(b) Constraint regions (blue).

(c) Isometric view.

Fig. 3. The initial geometry for the L-bracket problem, with all boundary conditions and constraint regions labelled.

(a) Iteration 0 (b) Iteration 5 (c) Iteration 25 (d) Iteration 100 (e) Iteration 275

Fig. 4. The L-bracket at various iterations during the optimisation.

(a) (b) (c)

Fig. 5. The L-bracket after optimisation. Subplot (a) overlays the initial geometry in green.

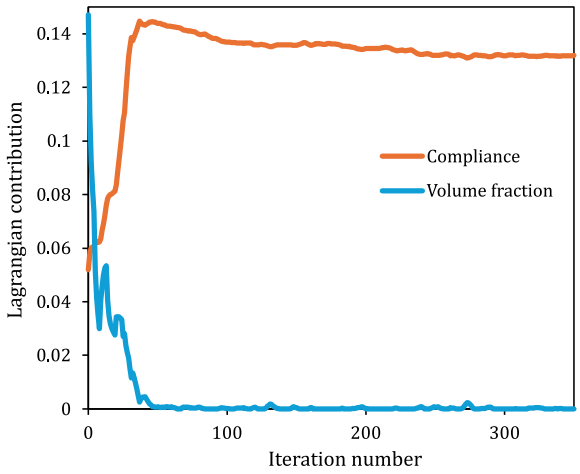


Fig. 6. Convergence plot showing the different contributions to the Lagrangian for the L-bracket. The values are smoothed by a moving average with a window size of 5. Note that the contribution of the volume constraint converges to zero, indicating that the constraint is satisfied.

and we expect that there is an equally optimal shape that can be found by reflecting the shape in Fig. 5 about the plane defined by the applied force vector and the pin-hole axis.

4.3.3. Cantilever

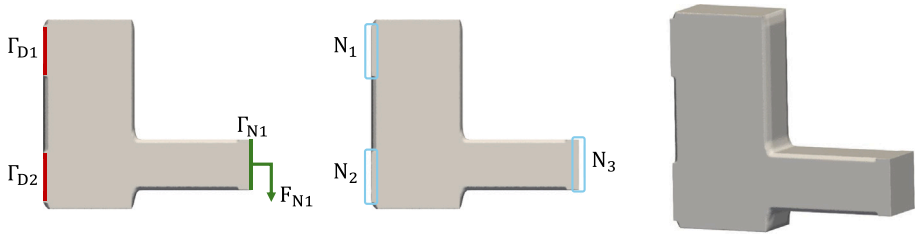
The initial geometry of the cantilever is shown in Fig. 7. A downward load is applied to the face indicated by the green line, and fixed boundary conditions are imposed on the red lines. The blue contours indicate constraint regions, with region N_3 being allowed to translate vertically, while regions N_1 and N_2 are frozen (i.e., are not allowed to change). Notice that the initial geometry and position of the loaded region are deliberately asymmetric. We minimise the compliance of the shape under the defined load, with a volume constraint ensuring that the volume of our final component does not exceed 30 % of the initial volume. For this problem with a characteristic length of 80, the smoothing parameter γ^2 was set to 4000. This is comparatively larger than the smoothing parameter for the other two problems; we chose γ^2 to be large to help avoid the uncontrolled hole generation/removal discussed in the following paragraph.

The evolution of the shape throughout the optimisation is shown in Fig. 8, and the final shape is shown in Fig. 9. Fig. 10 shows the convergence of the Lagrangian for this problem. This problem demonstrates our ability to optimise for the load interface location (region N_3) and also demonstrates that we can reproduce a theoretical cantilever result: a Michell truss, albeit a very simple one, similar to those found in [1,12,28]. We make a cautionary note here: the changes in topology (i.e., the addition/removal of holes) are due to numerical dissipation, as discussed in Section 4.3.1, and thus we do not have suitable methods of controlling these topology changes. This uncontrolled hole generation/removal leads to slow convergence, which could explain the high number of iterations required here.

4.3.4. Vehicle suspension rocker

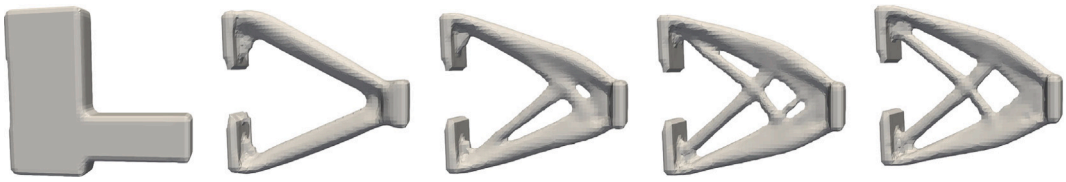
The initial geometry of the suspension rocker is shown in Fig. 11. Loads are applied to the green circles (representing pin joints that connect to the rest of the suspension assembly) in the direction of the green arrows with their relative magnitude represented by the arrow length. Note that the F_{N3} force has a component that is in the out-of-plane direction, which cannot be seen in Fig. 11. A fixed boundary condition is imposed on the red line. The blue contours indicate constraint regions, with region N_1 being allowed to translate in-plane, while regions N_2 and N_4 are frozen. Constraint region N_3 only allows for rotation of the region about an axis such that the interface (another pin joint) can align with the applied force (the axis of rotation and the direction of the force are more obvious in the results in Fig. 13). We again minimise the compliance of the shape under the defined load, with a volume constraint ensuring that the volume of our final component does not exceed 70 % of the initial volume. For this problem with a characteristic length of 175, the smoothing parameter γ^2 was set to 1000.

The evolution of the shape throughout the optimisation is shown in Fig. 12, and the final shape is shown in Fig. 13. Fig. 14 shows



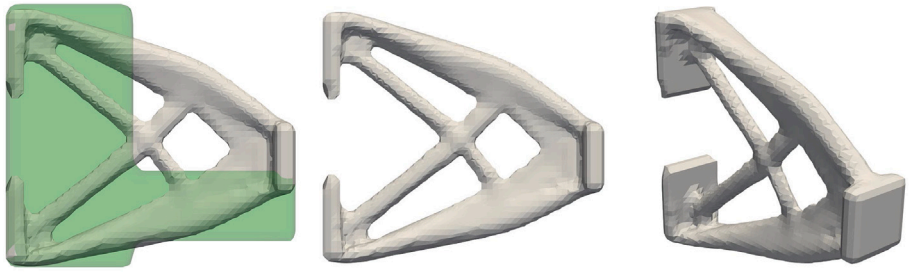
(a) Fixed interfaces (red), loaded interfaces (green) and applied loads. (b) Constraint regions (blue). (c) Isometric view.

Fig. 7. The initial geometry for the cantilever problem, with all boundary conditions and constraint regions labelled.



(a) Iteration 0 (b) Iteration 50 (c) Iteration 500 (d) Iteration 1000 (e) Iteration 1632

Fig. 8. The cantilever at various iterations during the optimisation.



(a) (b) (c)

Fig. 9. The cantilever after optimisation. Subplot (a) overlays the initial geometry in green.

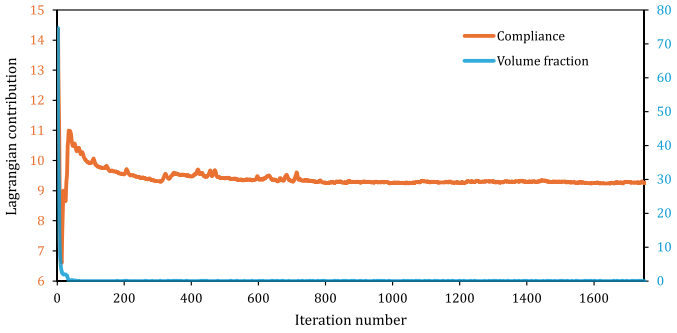


Fig. 10. Convergence plot showing the different contributions to the Lagrangian for the cantilever. The values are smoothed by a moving average with a window size of 5. Note the different scale for the y-axes of the two curves. Also, the contribution of the volume constraint converges to zero, indicating that the constraint is satisfied.

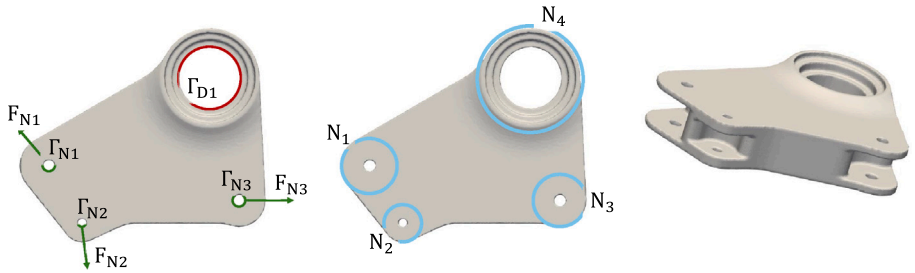
the convergence of the Lagrangian for this problem. This problem also demonstrates our ability to optimise interface regions, including their position (as in region N_1) and orientation (as in region N_3). In particular, the constraint region N_3 rotated to more-optimally align with the applied force, as depicted in Fig. 13.

5. Limitations and future work

The applications presented in the previous section show that our level set update procedure can be used within a level set-based topology algorithm to generate an optimal shape with interface regions having optimal location, orientation, and scale. While carrying out this work, a number of limitations have become apparent, some of which we now highlight.

Topology change. As discussed in earlier sections of the text, the level set update transport Eq. (1) cannot change the topology of the freely evolving part of the shape boundary in theory. However, since topology does change in practice due to numerical dissipation, this does not seem to be a practical limitation in the examples we have studied. Nevertheless, we have proposed a theoretical solution to this problem in Section 3.5 which we intend to investigate in future work.

Fine control. We have found instances where the behaviour of the evolving shape near the interfaces can be quite delicate, leading to a lack of robustness in the convergence of the topology optimisation procedure. For example, in some cases it can be difficult to prevent the evolving



(a) Fixed interfaces (red), loaded interfaces (green) and applied loads. (b) Constraint regions (blue). (c) Isometric view.

Fig. 11. The initial geometry for the rocker problem, with all boundary conditions and constraint regions labelled.

(a) Iteration 0 (b) Iteration 5 (c) Iteration 25 (d) Iteration 100 (e) Iteration 350

Fig. 12. The suspension rocker at various iterations during the optimisation.

(a) (b) (c)

Fig. 13. The suspension rocker after optimisation. The green arrow represents the direction of the applied force acting on Γ_{N1}. Subplot (a) overlays the initial geometry in green.

constrained region from merging into the evolving geometry in the free region or into another evolving constrained region, when this is the behaviour that is "desired" by the sensitivity analysis. Further research is needed to determine the most lightweight and graceful way of automatically detecting and handling situations such as these when they arise.

Smoothness of the evolving geometry. It is sometimes possible to observe a kink (a.k.a., discontinuity in the tangent planes) arise and grow in the evolving shape boundary at the location where its constrained regions overlap with its free regions. This occurs because the C-HSE procedure solves a second-order PDE for the update velocity. We essentially prescribe Dirichlet boundary conditions where the constrained and free regions overlap, thereby ensuring continuity of the update velocity across the overlap. Consequently, we cannot expect the derivative of the update velocity across the overlap to be continuous as well. In practice, however, this effect is small and usually most apparent in the earlier iterations of topology optimisation. Later iterations tend to cause the geometry to become smoother as it converges towards local optimality. Nevertheless, it is possible to overcome this problem fully in theory by considering a stronger smoothing inner product in the HSE formulation. For example, we could consider a second-order smoothing energy that penalises the L^2 norm of the HSE field along with its first two derivatives. In this case, the associated fourth-order PDE allows both the update velocity and its first derivatives to be continuous across the overlap between the constrained and free regions. Such an approach is investigated in [26], where it is implemented in a lightweight manner using a mixed finite element method.

Computation time. Both the standard HSE procedure and our modification, the C-HSE procedure, require that an additional PDE be solved for the update velocity in each iteration of topology optimisation. These procedures can thus become burdensome if their computation time nears that of other computationally heavy steps in the topology optimisation method, e.g., the solution of the linear elasticity equation used to carry out the sensitivity analysis. However, these PDEs are typically solved in a much smaller domain than the elasticity equation (in a narrowband of the shape boundary rather than in the shape itself). In the case of the standard HSE procedure, which solves a scalar PDE in the narrowband, the computational expense is deemed acceptable. The C-HSE procedure, on the other hand, solves a vector PDE in a large subset of the narrowband and can therefore require up to three times as many degrees of freedom as the standard HSE procedure. The system matrix of the C-HSE equation is thus up to three times larger, making the computation time

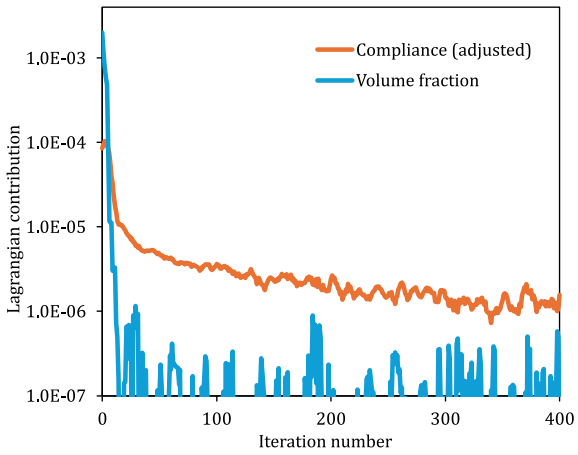


Fig. 14. Convergence plot showing the different contributions to the Lagrangian for the rocker. The "adjusted" compliance contribution is computed by subtracting the minimum value of the compliance Lagrangian contribution. The values are then smoothed by a moving average with a window size of 5. Note the use of a logarithmic scale for the y-axis. As with the other examples, the contribution of the volume constraint converges to zero, indicating that the constraint is satisfied.

correspondingly longer. Despite its increased size, we find that the computational expense of the elasticity equation remains the bottleneck of the topology optimisation method. We leave the investigation of methods for accelerating the solution procedure of the C-HSE equations to future work.

Smoothing parameter selection and update. From our observations on the effect of γ on the optimisation (see Section 4.3.1), we expect that a non-constant γ value could prove beneficial. Thus, as an avenue of future work, we look to develop an automated γ update procedure, in order to achieve fast convergence and lower final objective function values. We would also benefit from an automated way of selecting the (initial) value of γ.

Interface sensitivities. As previously discussed, in this work, we employ finite differencing to compute the shape gradient on the interfaces (i.e., the $G^{(n)}$ terms). In future work, we plan to explore a volumetric sensitivity analysis [9] to solve for these shape gradients more efficiently.

Extension to assemblies. In addition to transcending the limitations listed above, a more significant body of future work concerns the extension and application of our work to the multi-component setting. We remind the reader that the inspiration and intended application of our velocity update is for the co-design of one or more structural components of an assembly of components, such as the vehicle suspension system of Fig. 1. To this end, it will be necessary to add the following ingredients to our methodology. First, we must represent multiple designable components using separate level set functions that align appropriately along their interfaces. Second, we must compute sensitivity information from a full-assembly simulation, where load cases are prescribed at the external interfaces of the assembly (e.g., the tire and the chassis for the vehicle suspension system) and propagated throughout the assembly as design-dependent reaction loads at the interfaces between the assembly components. This new sensitivity formula will have to replace the right-hand side of our C-HSE method. Once this is done, the C-HSE update velocity can be used to update all level set functions simultaneously in a conflict-free manner since the integrated velocity field acts like a one-parameter family of diffeomorphisms of the background domain containing all geometry.

6. Conclusion

In this paper, we have developed a level set-based shape update procedure for shapes whose boundaries contain interfaces that must preserve their nature in the course of topology optimisation. We achieve this by incorporating translation, rotation, and scaling constraints in the interface regions into a novel HSE procedure for the level set update velocity in each iteration of the topology optimisation method. This procedure yields an interface-preserving update velocity that is a descent direction for the optimisation objective function. Consequently, the free-form part of the shape boundary, as well as the location, orientation, and scale of the interface regions can be optimised.

Finally, we note that, as with most topology optimisation formulations, the solutions obtained with the proposed method are not guaranteed to be unique. Multiple locally optimal configurations may satisfy the objective and constraints, and the final design can depend on factors such as the initial geometry and the smoothing parameter. The focus of this paper has been to introduce and demonstrate the C-HSE framework; a comprehensive exploration of the solution space, including the use of multiple initialisations or stochastic search strategies, is a valuable direction for future work.

CRediT authorship contribution statement

Adrian Humphry: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Mehran Ebrahimi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Nigel Morris:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Adrian Butscher:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Weak form of the C-HSE equations

By equating the variation of the objective in (13) to zero for all variations $\delta\Theta \in H_0^1(B', \mathbb{R}^3)$ and all variations $\delta z_s^{(n)} \in \mathbb{R}$, we obtain the weak form of the C-HSE equations:

$$\begin{aligned} 0 = &\int_{B'} \Theta \cdot \delta\Theta + \gamma^2 \int_{B'} D\Theta : D\delta\Theta \\ &+ \sum_{n=1}^{N} \sum_{s,t=1}^{d_n} \delta z_s^{(n)} z_t^{(n)} \int_{\mathcal{N}_I^{(n)}} \left(V_s^{(n)} \cdot V_t^{(n)} + \gamma^2 DV_s^{(n)} : DV_t^{(n)} \right) \\ &- \sum_{n=1}^{N} \int_{\partial\mathcal{N}_I^{(n)}} \frac{\partial\delta\Theta}{\partial n} \cdot \left(\Theta - \sum_{s=1}^{d_n} z_s^{(n)} V_s^{(n)} \right) \\ &- \sum_{n=1}^{N} \int_{\partial\mathcal{N}_I^{(n)}} \frac{\partial\Theta}{\partial n} \cdot \left(\delta\Theta - \sum_{s=1}^{d_n} \delta z_s^{(n)} V_s^{(n)} \right) \\ &- \int_{\partial\Omega\cap B'} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot \delta\Theta - \sum_{n=1}^{N} \sum_{s=1}^{d_n} \delta z_s^{(n)} \int_{\partial\Omega\cap\mathcal{N}_I^{(n)}} d\mathcal{J}_\Omega \, n_{\partial\Omega} \cdot V_s^{(n)}. \end{aligned} \tag{A.19}$$

Appendix B. Numerical implementation of the level set update

Constrained Hilbert space extension. The C-HSE procedure can be implemented quite straightforwardly using a finite element approach. First,

we describe the discrete versions of $\mathcal{B}$ and its relevant subsets. For simplicity, we will use the same notation as before for these objects. In every iteration of topology optimisation, we use the level set function of the current shape Ω to select a narrowband $\mathcal{B}$ of a fixed small size surrounding $\partial\Omega$ consisting of cells from the same uniform background grid used for the discretisation of the level set function and transport equation. We excise small internal neighbourhoods $\mathcal{B}$ consisting of all grid cells sufficiently close to each interface region $\Gamma_I^{(n)}$. We denote these by $\mathcal{N}_I^{(n)}$, and we denote the remaining region by $\mathcal{B}'$. The cell boundaries belonging to $\partial\mathcal{B}'$ adjacent to the non-design region are denoted by Γ_0.

Next, we introduce finite element shape functions $\{\phi_i : i \in nodes(\mathcal{B}')\}$ for all cells in $\mathcal{B}'$. Here, we use the notation $nodes(*)$ for the set of nodes contained in any simplicial subset of $\mathcal{B}'$. Without loss of generality, we can assume multilinear shape functions so that nodes coincide with the vertices of the background grid. Let e_α for $\alpha = 1, 2, 3$ be the standard basis vectors and let x_i for $i \in nodes(\mathcal{B}')$ be the node locations. We now introduce a finite element representation for the extension velocity in $\mathcal{B}'$ that explicitly includes the allowed motion constraints on the $\partial\mathcal{N}_I^{(n)}$ and the Dirichlet condition on Γ_0. We propose

$$\Theta := \sum_{i \in nodes(\mathcal{B}' \setminus \partial\mathcal{B}')} \sum_{\alpha=1}^{3} \Theta_{i\alpha} \phi_i e_\alpha + \sum_{n=1}^{N} \sum_{s=1}^{d_n} \left(\sum_{i \in nodes(\partial\mathcal{N}_I^{(n)})} \sum_{\alpha=1}^{3} z_s^{(n)} [V_s^{(n)}(x_i)]_\alpha \phi_i e_\alpha \right). \tag{B.20}$$

Finally, we derive the discrete version of the weak Eqs. (A.19) that we must solve for the coefficients $\Theta_{i\alpha}$ and $z_s^{(n)}$. To do this, we insert the representation (B.20) for both Θ and $\delta\Theta$ into (A.19) (or rather, its precursor

$$\langle\!\langle \Theta, \delta\Theta \rangle\!\rangle_\gamma = \int_{\partial\Omega} d\mathcal{J}_\Omega n_{\partial\Omega} \cdot \delta\Theta,$$

which holds when both Θ and $\delta\Theta$ satisfy the allowed motion constraints.) After some work, the result is a positive-definite system of linear equations. To express these equations compactly, we introduce the following notation. For each $\alpha = 1, 2, 3$ let $\Theta_\alpha := [\dots \Theta_{i\alpha} \dots]^\top \in \mathbb{R}^I$ where I is the number of nodes in $\mathcal{B}' \setminus \partial\mathcal{B}'$. For each $n = 1, \dots, N$ let $z^{(n)} := \left[\dots z_s^{(n)} \dots\right]^\top \in \mathbb{R}^{d_n}$ where $I^{(n)}$ is the number of nodes in $\partial\mathcal{N}_I^{(n)}$. Let

$$K_{ij} := \int_{\mathcal{B}' \setminus \partial\mathcal{B}'} \left(\phi_i \phi_j + \gamma \nabla\phi_i \cdot \nabla\phi_j \right)$$

be the relevant integrals of the finite element shape functions. We partition this data into a collection of matrices:

$$K^{(00)} := \begin{bmatrix} \vdots & \\ \cdots & K_{ij} \text{for } i,j \in nodes(\mathcal{B}' \setminus \partial\mathcal{B}') \quad \cdots \\ \vdots & \end{bmatrix} \in \mathbb{R}^{I \times I},$$

$$K^{(0\,n)} := \begin{bmatrix} \vdots & \\ \cdots & K_{ij} \text{for } i \in nodes(\mathcal{B}' \setminus \partial\mathcal{B}') \text{and } j \in nodes(\partial\mathcal{N}_I^{(n)}) \quad \cdots \\ \vdots & \end{bmatrix} \in \mathbb{R}^{I \times I^{(n)}},$$

$$K^{(nn)} := \begin{bmatrix} \vdots & \\ \cdots & K_{ij} \text{for } i,j \in nodes(\partial\mathcal{N}_I^{(n)}) \quad \cdots \\ \vdots & \end{bmatrix} \in \mathbb{R}^{I^{(n)} \times I^{(n)}}.$$

Note that there are no matrices of the form $K^{(nn')}$ for $n \neq n'$ since the interface regions are disjoint. Define the matrices of nodal values of the allowed motion fields

$$V_\alpha^{(n)} := \begin{bmatrix} \vdots & \\ \cdots & V_s^{(n)}(x_i) \text{for } i \in nodes(\partial\mathcal{N}_I^{(n)}) \text{and } s = 1, \dots, d_n \quad \cdots \\ \vdots & \end{bmatrix} \in \mathbb{R}^{I^{(n)} \times d_n}.$$

Finally, define the source terms

$$G_\alpha^{(0)} := \begin{bmatrix} \cdots & \int_{\partial\Omega} d\mathcal{J}_\Omega [n_{\partial\Omega} \cdot e_\alpha \, \phi_i \text{for } i \in nodes(\mathcal{B}' \setminus \partial\mathcal{B}') & \cdots \end{bmatrix}^\top \in \mathbb{R}^I,$$

$$G^{(n)} := \begin{bmatrix} \cdots & \int_{\partial\Omega} d\mathcal{J}_\Omega n_{\partial\Omega} \cdot V_s^{(n)} \text{for } s = 1, \dots, d_n & \cdots \end{bmatrix}^\top \in \mathbb{R}^{d_n}. \tag{B.21}$$

We can now express the discrete version of the weak Eqs. (A.19) that we must solve for the nodal vectors Θ_α and the z-coefficients in the following block-matrix form:

$$\begin{aligned} K^{(00)}\Theta_\alpha + \sum_n K^{(0\,n)} V_\alpha^{(n)} z^{(n)} &= G_\alpha^{(0)} && \text{for } \alpha = 1, 2, 3 \\ \sum_\alpha \left[V_\alpha^{(n)}\right]^\top \left[K^{(0\,n)}\right]^\top \Theta_\alpha + \sum_\alpha \left[V_\alpha^{(n)}\right]^\top K^{(nn)} V_\alpha^{(n)} z^{(n)} &= G^{(n)} && \text{for } n = 1, \dots, N. \end{aligned} \tag{B.22}$$

The system of Eqs. (B.22) can be assembled using standard finite-element methods (e.g., element-wise integration of shape functions and source terms), and it can be solved using any standard linear solver. We note that the total number of degrees of freedom of this system is equal to $3I + \sum_{n=1}^{N} d_n$, which is typically significantly smaller than the number of degrees of freedom of the finite element discretisation used for the elasticity equations in the shape Ω since $\mathcal{B}$ is a small narrowband of $\partial\Omega$.

Transport equation. We implement the time-dependent transport equation for the update of the level set function using the straightforward first-order upwind finite-difference scheme described in [24] on a fixed background grid. We simply modify this scheme by updating the velocity at each time step according to (16). In principle, one might want to improve the scheme to higher-order accuracy in order to prevent unwanted numerical drifting of the solution in the constrained regions, ensuring that the allowed motion occurs exactly. However, we have found that first-order accuracy is sufficient for our purposes.

Data availability

No data was used for the research described in the article.

References

[1] Alacoque L, James KA. Topology optimization with variable loads and supports using a super-Gaussian projection function. Struct Multidiscip Optim 2022; 65(2):50.

[2] Allaire G, Dapogny C, Estevez R, Faure A, Michailidis G. Structural optimization under overhang constraints imposed by additive manufacturing technologies. J Comput Phys 2017;351:295–328.

[3] Allaire G, Dapogny C, Jouve F. Shape and topology optimization. In: Handbook of numerical analysis. Vol. 22: Elsevier; 2021. pp. 1–132.

[4] Allaire G, Jouve F, Toader A-M. Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 2004;194(1):363–93.

[5] Allaire G, Jouve F, Michailidis G. Thickness control in structural optimization via a level set method. Struct Multidiscip Optim 2016;53.

[6] Ambrozkiewicz O, Kriegesmann B. Simultaneous topology and fastener layout optimization of assemblies considering joint failure. Int J Numer Methods Eng 2001;122(1):294–319.

[7] Bruno L. geogram. https://github.com/BrunoLevy/geogram

[8] Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. Comput Methods Appl Mech Eng 2001;190(26–27):3443–59.

[9] Dapogny C, Lebbe N, Oudet E. Optimization of the shape of regions supporting boundary conditions. Numer Math 2020;146(1):51–104.

[10] De Gournay F. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. SIAM J Control Optim 2006;45(1):343–67.

[11] Delfour MC, Zolésio J-P. Shapes and geometries: metrics, analysis, differential calculus, and optimization. SIAM; 2011.

[12] Dunning PD, Kim HA. Introducing the sequential linear programming level-set method for topology optimization. Struct Multidiscip Optim 2015;51: 631–43.

[13] Ebrahimi M, Behdinan K. A novel approach for design and optimization of automotive aluminum cross-car beam assemblies, Tech. report, SAE Technical Paper; 2015.

[14] Evans LC. Partial differential equations. Vol. 19. American Mathematical Society; 2022.
[15] Guo X, Zhang W, Zhong W. Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. J Appl Mech 2014;81(8):081009.
[16] Lee T-U, Xie Y. Simultaneously optimizing supports and topology in structural design. Finite Elem Anal Des 2021;197:103633.
[17] Morris NJW, Jayaraman PK, Butscher A. Bento: Beam network topology optimization. Comput Aided Des 2023;156:103439.
[18] Museth K, et al. OpenVDB. https://www.openvdb.org/about/
[19] Nitsche J. On Dirichlet problems using subspaces with nearly zero boundary conditions. In: The mathematical foundations of the finite element method with applications to partial differential equations. Elsevier; 1972. pp. 603–27.
[20] Nocedal J, Wright SJ. Numerical optimization. Springer; 1999.
[21] Norato JA, Bell BK, Tortorelli DA. A geometry projection method for continuum-based topology optimization with discrete elements. Comput Methods Appl Mech Eng 2015;293:306–27.
[22] Rakotondrainibe L, Allaire G, Orval P. Topology optimization of connections in mechanical systems. Struct Multidiscip Optim 2020;61(6):2253–69.
[23] Seltmann S, Hasse A. Topology optimization of compliant mechanisms with distributed compliance (hinge-free compliant mechanisms) by using stiffness and adaptive volume constraints instead of stress constraints. Mech Mach Theory 2023;180:105133.
[24] Sethian JA, et al. Level set methods and fast marching methods. Vol. 98. Cambridge Cambridge UP; 1999.
[25] Spivak M. Calculus on manifolds. Addison Wesley; 1965.
[26] Stein O, Grinspun E, Wardetzky M, Jacobson A. Natural boundary conditions for smoothing in geometry processing. ACM Trans Graph 2018;37(2):1–13.
[27] Swartz KE, James KA. Gaussian layer connectivity parameterization: a new approach to topology optimization of multi-body mechanisms. Comput Aided Des 2019;115:42–51.
[28] Wang MY, Wang X, Guo D. A level set method for structural topology optimization. Comput Methods Appl Mech Eng 2003;192(1–2):227–46.
[29] Wegert ZJ, Roberts AP, Challis VJ. A Hilbertian projection method for constrained level set-based topology optimisation. Struct Multidiscip Optim 2023;66(9):204.
[30] Zhu B, Zhang X, Zhang H, Liang J, Zang H, Li H, et al. Design of compliant mechanisms using continuum topology optimization: a review. Mech Mach Theory 2020;143:103622.