Author version

Published as:

Fletcher, A. H., Smith, H. A., & Hwang, J. T. (2026). Efficient and robust enforcement of geometric non-interference constraints for large-scale multidisciplinary design optimization. In AIAA AVIATION 2026 Forum (Paper No. AIAA 2026-4501). https://doi.org/10.2514/6.2026-4501

BibTeX for this reference:

https://lsdolab.github.io/lsdo-publications/publication/fletcher2026efficient/

```
@inproceedings{fletcher2026efficient,
  author = {Andrew H. Fletcher and Hollis A. Smith and John T. Hwang},
  title = {Efficient and Robust Enforcement of Geometric Non-Interference Constraints for
    Large-Scale Multidisciplinary Design Optimization},
  booktitle = {AIAA AVIATION 2026 Forum},
  year = {2026},
  doi = {10.2514/6.2026-4501}
}
```

Efficient and Robust Enforcement of Geometric Non-interference Constraints for Large-Scale Multidisciplinary Design Optimization

Andrew H. Fletcher*
University of California, San Diego, La Jolla, CA, 92093

Hollis A. Smith†
Air Force Research Laboratory, Wright-Patterson AFB, OH, 45433, USA

John T. Hwang‡
University of California, San Diego, La Jolla, CA, 92093

Geometric non-interference constraints are essential for ensuring that optimized designs remain physically realizable, particularly in applications requiring precise packing, clearance, and component integration. When discretized, they can generate an intractable number of inequality constraints, thereby degrading the performance of gradient-based optimizers. This work introduces an efficient and robust method for enforcing geometric non-interference constraints for arbitrary geometries. The method leverages a partial aggregation approach to improve the scaling while limiting the effect of ill-conditioning introduced by aggregation techniques. Specifically, the method uses signed distance functions for each component to aggregate constraints across one component boundary per pair. These are then leveraged to create an approximate signed distance function of the smooth union of the components to reduce constraint scaling from quadratic to linear in the number of components. To improve robustness, a minimum clearance is enforced to bound component interpenetration between discretization points. Finally, local constraint aggregation is applied to decouple the interpenetration bound from the constraint count. The method is benchmarked on problems ranging from academic to large-scale multidisciplinary design optimization. One of the academic problems, a box packing problem designed to highlight the ill-conditioning introduced by constraint aggregation, is solved in 73% fewer major iterations and 77% less time compared to a full-aggregation approach. The method is also benchmarked on a large-scale aero-structural multidisciplinary design optimization of the blended-wing-body aircraft, converging to a physically realizable optimum in 36% fewer major iterations and 46% less time compared to a full-aggregation approach for the non-interference constraints. By improving the scalability and robustness of geometric non-interference constraint enforcement, this work enables greater geometric complexity when solving multidisciplinary design optimization problems.

I. Introduction

Multidisciplinary design optimization (MDO) has proven to be effective for improving engineering designs that involve coupled disciplines. Advancements across a variety of fields, particularly physics modeling, have enabled its application to increasingly complex systems. As the complexity and functional density of systems that MDO is applied to continues to grow, the ability to precisely formulate constraints is vital to ensuring the optimized systems are feasible with regard to the design intent. In particular, geometric non-interference constraints, which ensure that components do not spatially interfere or intersect, play a central role in satisfying packing, storage, and layout requirements [1, 2]. These constraints enable precise control over the spatial relationships between components, forcing optimized designs to respect functional and physical integration limits.

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, AIAA Student Member, afletche@ucsd.edu
†Research Aerospace Engineer, Multidisciplinary Science and Technology Center, AIAA Member, hollis.smith.ctr@us.af.mil
‡Associate Professor, Deparment of Mechanical and Aerospace Engineering, AIAA Senior Member, jhwang@ucsd.edu
Distribution A: Approved for public release; distribution is unlimited. AFRL-2026-2044

While many prior works have imposed volume constraints to achieve similar requirements [3–5], such formulations are often only surrogate representations of more detailed spatial constraints. In cases such as accommodating landing gear, actuators, batteries, or other subsystem payloads, it is more precise to directly constrain the geometries to prevent interference. By capturing the true geometric relationships, such formulations improve the likelihood that optimized designs are physically realizable for complex systems and parameterizations.

Despite their importance, geometric non-interference constraints remain challenging to enforce efficiently and accurately within MDO problems. A framework has been proposed for packing cylinders within a rectangular container [6]. However MDO problems often involve arbitrarily-shaped geometries. When enforcing geometric non-interference between multiple arbitrarily-shaped geometries, the resulting constraints are inherently functional, requiring an infinite number of constraints for exact enforcement. A common discretization-based approach samples candidate contact boundaries and enforces pairwise distance constraints between points and potentially edges or faces [7–9]. However, this method scales poorly with both the discretization density and the number of interacting components. Moreover, for any finite discretization, gaps remain between sample points, which can allow potentially unbounded geometric interference while still satisfying the constraint formulation. For coarse discretizations, these gaps are larger, leading to ineffective constraint enforcement. For finer discretizations, the poor scaling can lead to prohibitive computational costs in evaluating and storing constraint Jacobians [7]. Constraint aggregation techniques, such as passing all the constraints through the Kreisselmeier–Steinhauser (K–S) function [10], have been used to mitigate this scaling [2, 7], though they can cause poor convergence behavior when multiple constraints are active or near-active simultaneously [1].

One class of approaches replaces the double-sided discretization with a signed distance function (SDF) defined for one side of the contact [2, 11, 12]. Since an SDF returns specifically the minimum of the distance to an object, leveraging an SDF essentially aggregates along one side of the contact pairing, thereby reducing the number of constraints. Furthermore, since the distances are signed, interior and exterior relationships can be defined clearly without additional constraints. Additionally, the derivative of the signed distance across the object boundary is continuous. SDF-based approaches have been used in applications such as wind farm layout optimization [11] and trajectory optimization [13], but these studies have typically been limited to enforcing non-interference between a set of points and an object. In contrast, MDO problems often require geometric non-interference constraints between multiple objects with non-degenerate geometries. SDF-based approaches have also been commonly applied for enforcing contact in dynamic simulations [14–16]. In contact dynamics, the non-interference is enforced for each time step individually. Since these problems only span a small time step, the objects are typically restricted in their potential motion. As a result, such methods usually identify a relatively small set of potential contact points that remains fixed over each time step. Since MDO problems can feature very large motion or deformation of components over an optimization, these strategies are typically not as applicable for MDO problems.

The goal of this work is to advance the capability of MDO to handle detailed geometric constraints by introducing an efficient and robust approach to enforcing geometric non-interference. The proposed methodology identifies an efficient set of geometric non-interference constraints that reduces optimization time for a given enforcement accuracy. It leverages partial aggregation to improve scaling without substantially increasing the number of optimization iterations. Specifically, each component is discretized once and then, for each discretization point, the minimum signed distance to the union of all other components is evaluated, effectively aggregating across the boundaries of all other components. To improve robustness, a minimum distance is imposed at each discretization point, which guarantees a bound on potential component interpenetration. Finally, the method leverages local aggregation to exploit knowledge about the potential contacts to further improve the scaling.

The contribution of this work is the proposed novel methodology for enforcing geometric non-interference constraints efficiently within MDO problems and the corresponding numerical benchmarking. The benchmark cases range from purely academic problems that isolate the effects of aggregation to a full MDO of a blended-wing-body (BWB) aircraft.

II. Background

A. Constraint Aggregation

Constraint aggregation replaces a set of m scalar constraints $\{c_i(x) \le 0\}_{i=1}^{m}$ with a single aggregated constraint $c_{\text{agg}}(x) \le 0$. The fundamental requirement is *consistency*: the feasible set of the aggregated constraint must be a subset of the original feasible set,

$$\big\{x : c_{\text{agg}}(x) \le 0\big\} \;\subseteq\; \big\{x : c_i(x) \le 0,\ \forall\, i \in \{1, \ldots, m\}\big\}. \tag{1}$$

In general, the inclusion is strict, so c_{agg} is a conservative surrogate for the original constraint set.

The primary motivation for aggregation is computational. Evaluating the constraint Jacobian $\partial c/\partial x \in \mathbb{R}^{m\times n}$ via adjoint methods requires one back-propagation per constraint. Therefore, reducing from m constraints to one reduces the number of back-propagations to one. Furthermore the memory cost from the Jacobian storage decreases from $O(mn)$ to $O(n)$.

The cost of this reduction is that the aggregated Jacobian $\partial c_{\text{agg}}/\partial x \in \mathbb{R}^{1\times n}$ is a single row vector, providing the optimizer with a single sensitivity direction rather than m separate ones. Gradient-based optimizers suitable for large-scale MDO linearize constraints at each iteration, so each update step acts only on the information in this one row. Whether this leads to ill-conditioning depends on which individual sensitivities are represented in that row, and this is where aggregation approaches differ.

To illustrate the tradeoff, consider the degenerate case of summing all constraints, $c_{\text{sum}}(x) := \sum_{i=1}^{m} c_i(x)$. The Jacobian $\partial c_{\text{sum}}/\partial x = \sum_i \partial c_i/\partial x$ contains sensitivity to every constraint simultaneously. However, $c_{\text{sum}}(x) \le 0$ permits individual $c_i > 0$ provided other constraints are sufficiently negative, so Eq. (1) is violated and summation is not a valid aggregation method. It is included here only to motivate the tradeoff: including more individual sensitivities in the aggregated Jacobian prevents discontinuous switching but risks allowing the optimizer to trade feasibility of one constraint against another.

Smooth Maximum

The most widely used consistent aggregation approach replaces $\{c_i(x)\}_{i=1}^{m}$ with a smooth approximation to the maximum. Defining $c(x) := [c_1(x), \ldots, c_m(x)]^\top \in \mathbb{R}^m$,

$$c_{\text{agg}}(x;\rho) := \widetilde{\max}(c(x);\,\rho)\,, \tag{2}$$

where $\widetilde{\max}(\,\cdot\,;\rho) : \mathbb{R}^m \to \mathbb{R}$ is a smooth maximum function with sharpness parameter $\rho > 0$ that converges to $\max_i c_i(x)$ as $\rho \to \infty$. A common choice is the Kreisselmeier–Steinhauser (KS) function [10],

$$c_{\text{KS}}(x;\rho) := \frac{1}{\rho}\ln\left(\sum_{i=1}^{m} e^{\rho\, c_i(x)}\right), \tag{3}$$

which satisfies $c_{\text{KS}}(x;\rho) \ge \max_i c_i(x)$ for all $\rho > 0$, guaranteeing consistency (1). The overestimation decays as $O(\ln(m)/\rho)$, so larger ρ gives a tighter approximation at the cost of increased curvature. Other similar methods include using a p-norm or induced aggregation [17].

The aggregated Jacobian is dominated by the sensitivity of the most-violated constraint, with exponentially decaying contributions from the rest. Near a point where the most-active constraint switches, the Hessian $\nabla^2 c_{\text{KS}}$ becomes large, causing the first-order linearization used by the optimizer to become locally inaccurate. The optimizer therefore takes steps that improve the previously active constraint while potentially violating the newly active one. As a result, the number of optimization iterations required to converge often significantly increases. Throughout this work, regions where this linearization error is significant are referred to as *ill-conditioned regions*.

Maximum Rectifier Summation

An alternative consistent approach applies a rectifier to each constraint before summing [18],

$$c_{\text{rect}}(x) := \sum_{i=1}^{m} \max(c_i(x),\,0). \tag{4}$$

Since each term $\max(c_i, 0) \ge 0$, the sum satisfies $c_{\text{rect}}(x) \le 0$ if and only if $c_i(x) \le 0$ for all i, so consistency (1) holds with equality. Defining the violated index set $\mathcal{A}(x) := \{i \in \{1,\ldots,m\} : c_i(x) > 0\}$, the aggregated Jacobian is

$$\frac{\partial c_{\text{rect}}}{\partial x} = \sum_{i\in\mathcal{A}(x)} \frac{\partial c_i}{\partial x}, \tag{5}$$

which contains sensitivity primarily to all of the violated constraints and none of the satisfied ones. Two sources of ill-conditioning follow. First, transitions between active and inactive constraints cause near-discontinuous Jacobian rotations, as in the smooth-maximum approach. Second, when more than one constraint is violated, all violated constraint sensitivities enter the sum, thereby removing the near-discontinuous rotations but reintroducing the tradeoff behavior of the summation case but restricted to the violated subset.

B. Signed Distance Functions

Implicit geometry representations define object boundaries as level sets of scalar fields, rather than through explicit boundary or volumetric discretizations. The most common formulation is the signed distance function (SDF), which returns the signed distance to the closest point on the boundary, with the zero level set defining the surface, negative values indicating interior points, and positive values exterior points [19, 20]. SDFs are widely used in collision detection and simulation due to their ability to provide both distance and inside–outside information efficiently [19].

This work leverages SDF representations, in conjunction with explicit geometry, due to their suitability for defining one-sided non-interference constraints. In particular, SDF-based formulations implicitly aggregate constraint evaluations on one side of a contact interface, significantly reducing the number of constraints compared to pairwise checks between discretizations of both interacting components. However, constructing accurate and efficient SDFs for arbitrary geometries remains challenging.

Within SDF frameworks, three primary classes of construction methods are commonly used. First, primitive-based representations compose geometries from analytic shapes such as spheres, boxes, capsules, and ellipsoids, for which closed-form distance expressions exist [21, 22]. These primitives are typically combined using smooth minimum or blending operators to form a continuous field. Such approaches enable highly efficient and accurate distance evaluations, but the number of primitives required to represent complex geometries can grow significantly, increasing computational cost and introducing approximation error.

Second, fitting-based methods approximate the SDF using sampled distance data. These approaches construct surrogate models, such as spline-based [2] or radial-basis-function-based representations [23–27]. While they offer greater flexibility for representing arbitrary geometries, their accuracy depends on sampling density and can degrade in regions far from the boundary, particularly in the evaluation of gradients.

Third, projection-based methods evaluate the signed distance by directly computing the closest point on the explicit geometry during each query. When the projection operation is robust, these methods provide exact distance evaluations for arbitrary geometries, but at a higher computational cost than precomputed or fitted representations.

Because this work targets accurate and efficient constraint enforcement for arbitrarily complex geometries, accurate near-boundary distance evaluation is critical. Accordingly, the proposed approach uses primitive-based SDFs for geometries that admit analytic representations, and projection-based evaluation for more complex components. Fitting-based approaches remain a promising direction for future work due to their potential to accelerate evaluation while retaining geometric flexibility.

C. Functional Constraint Enforcement

The functional nature of geometric non-interference constraints implies an infinite set of pointwise conditions, requiring specialized treatment within optimization frameworks. An analogous challenge arises in the enforcement of boundary conditions in finite element methods, where two primary strategies are employed: strong and weak enforcement.

Strong enforcement directly constrains the solution space by eliminating degrees of freedom to satisfy boundary conditions exactly. However, this approach is generally impractical for geometric non-interference. Interacting bodies often reside in distinct function spaces, as they may be represented by different parameterizations or even possess different topologies, precluding a unified constraint elimination strategy. Additionally, because non-interference constraints are inequalities, strong enforcement would require dynamically removing and reintroducing degrees of freedom as constraints become active or inactive, which is incompatible with standard optimization workflows.

In contrast, weak enforcement incorporates constraints into the variational formulation through additional terms, such as penalty functions, Lagrange multipliers, augmented Lagrangian methods, or Nitsche-type formulations [28–30]. These approaches enforce constraints in an integral or variational sense rather than pointwise, allowing them to be applied without modifying the underlying function spaces. Although weak enforcement does not guarantee exact satisfaction at finite resolution, it integrates naturally with existing MDO formulations and converges to exact enforcement as the discretization is refined.

Framing geometric non-interference constraints within a weak enforcement paradigm therefore enables tractable discretization and aggregation strategies while remaining compatible with standard optimization methods. Accordingly, the proposed approach adopts and extends weak enforcement concepts to the treatment of geometric non-interference inequality constraints.

III. Mathematical Preliminaries

Geometric non-interference is enforced by requiring that the signed distance between every pair of boundary points belonging to distinct components be non-negative. The sign convention is chosen such that positive signed distance corresponds to exterior separation, so that feasibility requires $d_s > 0$. Without loss of generality, for notation, all non-interference constraints are treated as exterior relationships since interior constraints can be recovered by negating d_s. This formulation assumes that each component boundary is a compact, orientable manifold without boundary, which guarantees a well-defined interior–exterior partition and therefore a well-defined signed distance function for each component. This formulation further assumes that no component self-intersects.

To introduce notation formally, define:

- $f : \mathbb{R}^n \to \mathbb{R}$ as the objective function, where $x \in \mathbb{R}^n$ is the vector of design variables and n is the number of design variables,
- $c_e : \mathbb{R}^n \to \mathbb{R}^{m_e}$ as the equality constraint function with m_e equality constraints,
- $c_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ as the inequality constraint function for all constraints *excluding* geometric non-interference, with m_i constraints,
- $d_s : \mathbb{R}^{n_d} \times \mathbb{R}^{n_d} \to \mathbb{R}$ as the signed distance function between two points in $\mathbb{R}^{n_d}$, where n_d is the number of spatial dimensions and positive values indicate separation,
- $\hat{\Omega}_{jk} := [0,1]^{n_{d_r}}$ as the reference domain for the k-th boundary patch of the j-th component, where n_{d_r} is the parametric dimension and $\xi \in \hat{\Omega}_{jk}$ denotes a parametric coordinate,
- $\hat{\Omega}_j := \bigsqcup_{k=1}^{n_{k_j}} \hat{\Omega}_{jk}$ as the disjoint union of reference domains covering the full boundary of component j, where n_{k_j} is the number of boundary patches,
- $p_j : \mathbb{R}^n \to C^{0,1}(\hat{\Omega}_j, \mathbb{R}^{n_d})$ as the boundary parametrization map, where $C^{0,1}(\hat{\Omega}_j, \mathbb{R}^{n_d})$ denotes the space of Lipschitz-continuous maps from $\hat{\Omega}_j$ to $\mathbb{R}^{n_d}$, so that for fixed x the function $\xi \mapsto p_j(x)(\xi)$ parametrizes the boundary of component j,
- $\partial\Omega_j(x) := \{\, p_j(x)(\xi) \mid \xi \in \hat{\Omega}_j \,\} \subset \mathbb{R}^{n_d}$ as the resulting boundary of component j in physical space, and
- $c_g : \mathbb{R}^n \times \hat{\Omega}_{j_a} \times \hat{\Omega}_{j_b} \to \mathbb{R}$, defined by

$$c_g(x, \xi_a, \xi_b;\, j_a, j_b) := -d_s\big(p_{j_a}(x)(\xi_a),\, p_{j_b}(x)(\xi_b)\big)\,,$$

  as the geometric non-interference constraint function for a pair of boundary points on components j_a and j_b, respectively.

As such, the optimization with geometric non-interference constraints is formulated as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c_e(x) = \mathbf{0}, \\ & c_i(x) \le \mathbf{0}, \\ & c_g(x, \xi_a, \xi_b;\, j_a, j_b) \le 0 \\ & \qquad \forall\, (\xi_a, \xi_b) \in \hat{\Omega}_{j_a} \times \hat{\Omega}_{j_b},\ \forall\, j_a \neq j_b \in \{1, \ldots, n_c\}, \end{aligned} \tag{6}$$

where n_c is the number of components. The constraint $c_g \le 0$ enforces $d_s \ge 0$ for all boundary-point pairs across distinct components, preventing geometric interference. As such, the geometric non-interference are defined as a functional constraint and scales with $n_c(n_c - 1)$ where n_c is the number of components that can potentially make contact.

IV. Geometric Non-interference Constraint Enforcement Using Partial Aggregation

The partial aggregation approach introduces aggregation selectively over subsets of constraints in which only a single constraint is expected to be active at a time and any transition in the active constraint is smooth. As discussed in Section II.A, this selectivity avoids introducing ill-conditioning into regions the optimizer is likely explore while still decreasing the scaling of the number of optimization constraints.

Before presenting the method, it is useful to distinguish between two different classes of non-interference constraints. An exterior constraint requires that one component lie outside another, while an interior constraint requires that one component lie inside another. Notably, the two classes differ in the convexity structure of their feasible sets. The feasible set of an exterior constraint is the complement of the interior of the enclosing component, which is non-convex in general, even when the enclosing component is convex. This non-convexity can introduce local optima into the

optimization landscape. In contrast, the feasible set of an interior constraint is the interior of the enclosing component, which is convex whenever the enclosing component is convex, and connected for any simply connected enclosing component. Interior constraints therefore introduce fewer local optima and are generally better conditioned than exterior constraints. This distinction motivates a methodological choice in the proposed method: for component pairs in which one component must lie inside another, the discretization is placed on the interior component and evaluated through the SDF of the enclosing component without negation. The enclosing component acts solely as an SDF target and is never discretized for this constraint pair. This asymmetric enforcement exploits the better-conditioned structure of the interior constraint while avoiding the additional discretization points that symmetric enforcement would require. In contrast, exterior-exterior relationships are enforced in both directions to reduce the likelihood of interpenetration and prevent one component from enclosing the other.

The method is presented in four steps. First, an SDF for each component is introduced, which exactly aggregates the non-interference constraints across the boundary of one component per pair. Second, a smooth union SDF over all other components approximates the exact union, reducing the constraint count from $O(n_c^2)$ to $O(n_c)$. Third, a minimum clearance distance is enforced at each discretization point to bound interpenetration between points. Finally, local aggregation is applied to decouple the constraint count from the discretization resolution.

A. Constraint Aggregation Using Signed Distance Functions

The continuous constraint formulation (Eq. 6) poses two computational challenges: the geometric non-interference constraint c_g is a functional constraint over a continuous domain, and evaluating the signed distance between two arbitrary boundary points as formulated is not clearly defined. Both challenges are addressed by introducing a signed distance function (SDF) for each component. Let

$$d_{s_j} : \mathbb{R}^{n_d} \to \mathbb{R} \tag{7}$$

denote the SDF of the j-th component, where $d_{s_j}(p) > 0$ when p is exterior to component j and $d_{s_j}(p) < 0$ when p is interior. By definition,

$$d_{s_j}(p) = \min_{\xi \in \hat{\Omega}_j} d_s\big(p,\, p_j(x)(\xi)\big)\,, \tag{8}$$

so d_{s_j} aggregates the signed distance over the entire boundary of component j. Substituting into the continuous constraint formulation, the geometric non-interference constraint function reduces to

$$c_g(x, \xi;\, j_a, j_b) := -d_{s_{j_b}}\big(p_{j_a}(x)(\xi)\big)\,, \quad c_g : \mathbb{R}^n \times \hat{\Omega}_{j_a} \to \mathbb{R}, \tag{9}$$

where boundary points of component j_a are evaluated through the SDF of component j_b. The constraint domain is then made finite by discretizing $\hat{\Omega}_j$ with n_{I_j} points $\{\xi_I\}_{I=1}^{n_{I_j}}$, yielding the discretized constraint function

$$c_g(x, \xi_I;\, j_a, j_b) := -d_{s_{j_b}}\big(p_{j_a}(x)(\xi_I)\big)\,, \quad c_g : \mathbb{R}^n \times \{1, \ldots, n_{I_{j_a}}\} \to \mathbb{R}, \tag{10}$$

and the discretized non-interference constraints

$$\begin{aligned} &c_g(x, \xi_I;\, j_a, j_b) \le 0 \\ &\qquad \forall\, I \in \{1, \ldots, n_{I_{j_a}}\},\ \forall\, j_a \neq j_b \in \{1, \ldots, n_c\}, \end{aligned} \tag{11}$$

with a total constraint count that scales as $O(n_I \cdot n_c^2)$, where n_I denotes a representative discretization size.

The key assumption underlying this aggregation is that $d_{s_{j_b}}$ varies smoothly in the neighborhood of each active discretization point. Two geometric conditions can violate this assumption and introduce ill-conditioning. First, $d_{s_{j_b}}$ is non-smooth at points equidistant from two or more distinct features of $\partial\Omega_{j_b}$, where the closest feature switches discontinuously and the gradient of $d_{s_{j_b}}$ is undefined. Second, even when the closest point is unique, regions of high boundary curvature κ cause the gradient of $d_{s_{j_b}}$ to rotate rapidly over distances of $O(1/\kappa)$, degrading the accuracy of the linear constraint approximation used by the optimizer within a single step. Both conditions are unlikely to cause significant ill-conditioning in practice for sufficiently fine discretizations, particularly since components are assumed not to self-intersect. As such, this work assumes that the reduction in constraint count outweighs the risk of ill-conditioning.

For geometrically simple components, analytic SDFs are used as they are fast and accurate. For components with complex geometry, a projection-based SDF is used, which computes the signed distance via a closest-point projection onto the explicit geometry representation. The projection-based SDF is more expensive but applicable to arbitrary geometries and is perfectly accurate as long as the projection converges robustly and accurately. An evaluation of a projection-based SDF along four cross sections is shown in Figure 1.

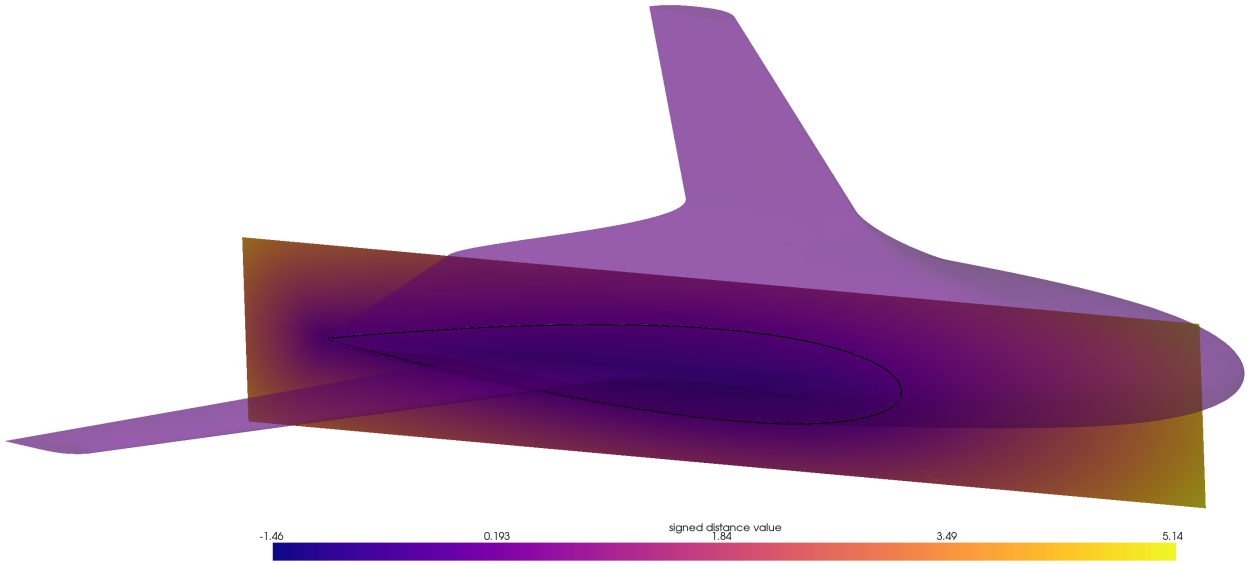


Fig. 1 An evaluation of a projection-based signed distance function along a cross-section plane for a blended-wing-body aircraft is plotted as well as the corresponded blended-wing-body aircraft. The color corresponds to the signed distance for that point in space.

B. Constraint Aggregation Using Signed Distance Functions of Smooth Union of Components

The $O(n_c^2)$ constraint scaling is reduced to $O(n_c)$ by replacing the per-pair SDF evaluations with an SDF of the smooth union of all other components. The exact union of a set of SDFs is given by the pointwise minimum of their outputs [20],

$$d_{s_{-j}}(p) := \min_{b \neq j} d_{s_b}(p), \quad p \in \mathbb{R}^{n_d}, \tag{12}$$

which satisfies $d_{s_{-j}}(p) \geq 0$ if and only if p is exterior to all components other than j. Because the hard minimum is non-smooth at points equidistant from two components, it is replaced by a smooth minimum,

$$\widetilde{d}_{s_{-j}}(p;\rho) := \widetilde{\min}\left(\{d_{s_b}(p)\}_{b \neq j};\, \rho\right), \tag{13}$$

where $\widetilde{\min}(\,\cdot\,;\rho) : \mathbb{R}^{n_e} \to \mathbb{R}$ is a smooth minimum function over n_e scalar inputs with sharpness parameter $\rho > 0$. As $\rho \to \infty$, $\widetilde{\min}$ converges to the exact minimum, so $\widetilde{d}_{s_{-j}}$ approximates $d_{s_{-j}}$ with error that decreases as ρ increases. The geometric non-interference constraint function then becomes

$$c_g(x, \xi_I;\, j) := -\widetilde{d}_{s_{-j}}\left(p_j(x)(\xi_I);\, \rho\right), \quad c_g : \mathbb{R}^n \times \{1, \ldots, n_{I_j}\} \to \mathbb{R}, \tag{14}$$

and the optimization constraints reduce to

$$\begin{aligned} &c_g(x, \xi_I;\, j) \leq 0\,, \\ &\qquad \forall\, I \in \{1, \ldots, n_{I_j}\},\ \forall\, j \in \{1, \ldots, n_c\}, \end{aligned} \tag{15}$$

with a total constraint count that scales as $O(n_I \cdot n_c)$.

The assumption underlying this aggregation is that, for each discretization point on component j, no more than one other component simultaneously approaches that point. This is the condition under which the smooth minimum in Eq. (13) does not introduce ill-conditioning: if two components are equidistant from a discretization point, the smooth minimum transitions between them, and if that transition is near-discontinuous, the ill-conditioning will likely increase the number of optimization iterations required. As with the per-pair SDF aggregation, this is unlikely for sufficiently fine discretizations given the non-self-intersection assumption, and this work assumes the scaling improvement outweighs the risk of ill-conditioning.

C. Bounding Component Interpenetration Using Minimum Clearance Enforcement

With a finite discretization, a component may penetrate another between discretization points even when all constraints in Eq. (15) are satisfied. To bound this interpenetration, each constraint is augmented with a minimum clearance $d_{\min,jI} > 0$, replacing the non-interference condition $\widetilde{d}_{s_{-j}}(p_j(x)(\xi_I);\rho) \geq 0$ with

$$\widetilde{d}_{s_{-j}}\left(p_j(x)(\xi_I);\rho\right) \geq d_{\min,jI} \quad \forall\, I,\ j, \tag{16}$$

or equivalently, $c_g(x,\xi_I;\ j) + d_{\min,jI} \leq 0$. Geometrically, this enforces a spherical exclusion zone of radius $d_{\min,jI}$ in physical space around the image $p_j(x)(\xi_I) \in \mathbb{R}^{n_d}$ of each discretization point. The union of these exclusion spheres covers $\partial\Omega_j(x)$ if and only if every point on the boundary lies within distance $d_{\min,jI}$ of at least one discretization point. A sufficient condition for coverage under a structured grid on each boundary patch is

$$d_{\min,jI} \geq \tfrac{1}{2} \max_{\text{diagonal neighbors } I'} \left\| p_j(x)(\xi_I) - p_j(x)(\xi_{I'}) \right\|, \tag{17}$$

where the maximum is taken over the diagonal neighbors of point I in the grid, and distances are measured in physical space $\mathbb{R}^{n_d}$. When this condition holds, the boundary of the union of exclusion spheres provides a conservative outer approximation of $\partial\Omega_j(x)$, bounding the maximum interpenetration depth.

This introduces a tradeoff between error in the approximation of $\partial\Omega_j$ and the number of constraint. For a reference domain of dimension n_{d_r}, scaling $d_{\min,jI}$ by $\frac{1}{\alpha}$ requires $\alpha^{n_{d_r}}$ times as many discretization points and therefore constraints. For the typical case of a three-dimensional geometry with a two-dimensional reference domain ($n_{d_r} = 2$), halving $d_{\min,jI}$ requires four times as many constraints.

The recommended workflow is: set $d_{\min}$ to the maximum allowable boundary approximation error, choose the coarsest structured grid such that Eq. (17) is satisfied for all I and j, and verify that the resulting constraint count is computationally feasible. If not, increase $d_{\min}$.

If the geometry and parameterization are such that unbounded component interpenetration between discretization points is precluded, for instance, when the discretization points are constrained to lie on a convex boundary, then the minimum clearance $d_{\min}$ may be set to zero. In such cases, omitting the minimum clearance eliminates both the boundary approximation error and the additional conservatism it introduces, and reduces the problem to the formulation of Eq. (15). Additionally, with the same knowledge, it is often advantageous to use a relatively coarse discretization to decrease the computational cost.

D. Decoupling Discretization Resolution and Constraint Count Using Local Aggregation

The constraint count and discretization resolution can be largely decoupled through local constraint aggregation. For a structured grid, the constraints within a neighborhood of physical radius $r > 0$ around each discretization point are aggregated using a smooth maximum, yielding one aggregated constraint per block of discretization points,

$$\widetilde{c}_g(x,\xi_L;\ j) := \widetilde{\max}\left(\{c_g(x,\xi_{L'};\ j)\}_{L'\in\mathcal{N}(L,r)};\ \rho\right) \leq 0, \tag{18}$$

where

$$\mathcal{N}(L,r) := \left\{L' : \left\| p_j(x)(\xi_L) - p_j(x)(\xi_{L'}) \right\| \leq r\right\} \tag{19}$$

is the index set of discretization points within physical distance r of point L. Note that $\mathcal{N}(L,r)$ depends on x through $p_j(x)$; in practice, it is evaluated at a nominal or fixed configuration to avoid recomputing the neighborhood at every iteration. The aggregated constraint in Eq. (18) is satisfied if and only if $c_g(x,\xi_{L'};\ j) \leq 0$ for all $L' \in \mathcal{N}(L,r)$ in the limit $\rho \to \infty$; for finite ρ, feasibility is approximately enforced with error controlled by ρ.

Holding r fixed and refining the discretization (increasing n_{I_j} and correspondingly decreasing $d_{\min}$) does not change the number of aggregated constraints, since the number of neighborhood centers remains equal to n_{L_j} while each neighborhood contains more points. For the decoupling to be valid, the minimum clearance must satisfy $d_{\min} \leq r$; otherwise, the exclusion spheres from adjacent neighborhood centers do not overlap, and the union of spheres may fail to cover $\partial\Omega_j(x)$.

This introduces two independent tradeoffs. The first is between constraint count and the risk of ill-conditioning. A larger r reduces the number of constraints but increases the likelihood that multiple distinct contact features fall within a single neighborhood, leading to near-discontinuous switches in the active constraint and ill-conditioning. The second tradeoff is between boundary approximation accuracy and computational cost per constraint evaluation. Refining the discretization decreases $d_{\min}$ and increases the number of SDF evaluations and the size of each local aggregation, but

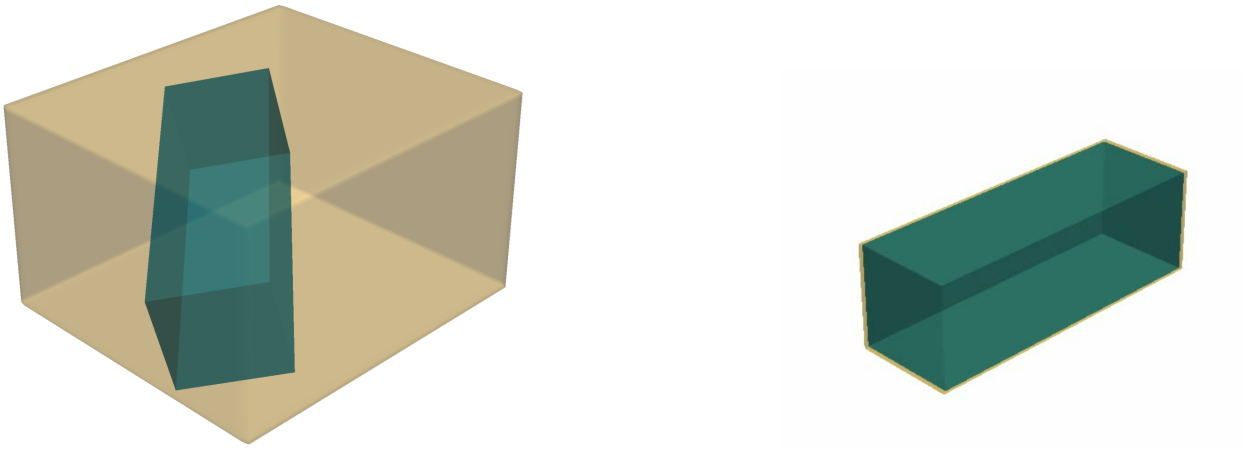
does not change the number of optimization constraints. In practice, SDF evaluations and smooth minimum operations are inexpensive, so the discretization can be refined substantially. However, when implemented within a graph-based automatic differentiation framework, each additional SDF evaluation enlarges the computational graph, increasing the cost of the model compilation stage.

The recommended workflow is: determine the largest aggregation radius r for which no ill-conditioning is expected, based on the geometry parameterization. Parameterizations with fine local shape control, many components, or components with smaller features generally require a smaller r. Then set $d_{\min} \leq r$ and choose the coarsest discretization satisfying both Eq. (17) and the coverage condition $d_{\min} \leq r$. If the resulting computational cost is too high, increase r at the cost of a higher risk of ill-conditioning or increase d_{min} while still ensuring $d_{\min} \leq r$. Enforcing $d_{\min} \leq r$ ensures at least one discretization point per local aggregation region when following this workflow.

V. Results

The method is benchmarked on four problems of increasing complexity. The first is a box packing problem designed to isolate the effect of constraint aggregation on optimizer convergence. The second is a multi cylinder packing problem with an armadillo-shaped obstacle, introducing multiple components and a complex geometry. The third is a geometric volume minimization of a BWB aircraft with multiple payload components, approaching a realistic case with many simultaneously interfering components and planform design variables. The fourth extends the BWB problem to a full aero-structural MDO.

A. Box Packing



(a) Initial condition with multiple active or near-active geometric non-interference constraints.

(b) Optimized solution in which the container contracts to the dimensions of the package box and the package box rotates and translates to match the container.

Fig. 2 The box packing problem is to minimize the container volume subject to the package box remaining interior to the container. The design variables include translation and rotation of the package box, as well as length, width, and height of the container.

The box packing problem features one package box that must remain interior to a container box, based on an academic packing benchmark from a report on MDO for packaging [1]. The package box has rigid translation and rotation design variables and the container box has length, width, and height as design variables. The objective is to minimize the container volume $V(x) = l \cdot w \cdot h$. The expected solution is for the package box to align with the container, and for the container to contract to the package dimensions (Figure 2). Since the package box must lie interior to the container, the non-interference constraint uses the container SDF directly without negation, so that feasibility corresponds to $d_{s_c}(p_j(x)(\xi_I)) \leq 0$. The discretization consists of the corners, face centers, and the four parametric points at coordinates $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$, and $(0.75, 0.75)$ on each face of the package box. The minimum clearance enforcement is intentionally omitted to isolate the effect of aggregation on convergence. The

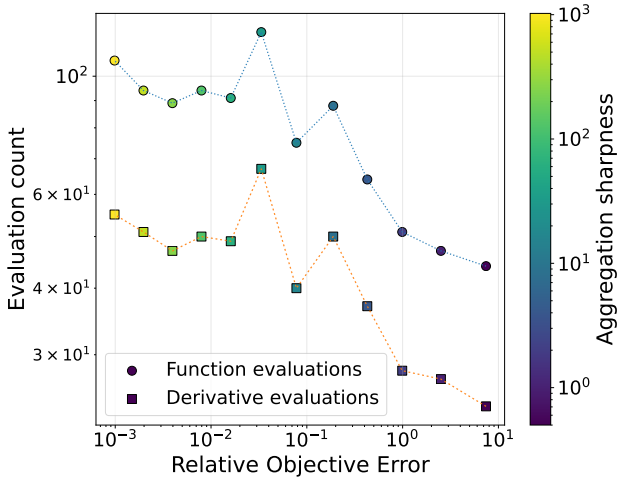


Fig. 3 A sweep over aggregation sharpness values is performed for the box packing problem with a full aggregation formulation. Higher sharpness, ρ (marker color), reduces the relative objective error at convergence but tends to increase the number of function evaluations (circles, blue) and derivative evaluations (squares, orange).

problem is formulated as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & V(x) \\ \text{s.t.} \quad & d_{s_c}\left(p_j(x)(\xi_I)\right) \leq 0 \quad \forall\, I \in \{1, \ldots, n_I\}, \end{aligned} \tag{20}$$

where d_{s_c} is the SDF of the container box and $\{\xi_I\}_{I=1}^{n_I}$ are the parametric coordinates of the discretization points on the package box boundary.

Using PySLSQP [31, 32] within modOpt [33], the proposed method converged in 15 major iterations and 19 function evaluations. To assess the effect of full aggregation, the problem is solved repeatedly over a sweep of aggregation sharpness values ρ, collapsing all geometric non-interference constraints into a single scalar constraint (Figure 3).

To achieve a relative objective error within 0.001, the sharpness must be increased to $\rho = 1024$. At this sharpness, full aggregation requires 55 major iterations and 107 function evaluations. Correspondingly, the proposed method requires 73% fewer major iterations, 82% fewer function evaluations, and 77% less total optimization time than a sufficiently accurate fully aggregated method.

To explore the tradeoff between enforcement accuracy and computational cost introduced by the minimum clearance, the full method is implemented following the workflow in Section IV.D and solved repeatedly over a sweep of minimum clearance values, both with and without local constraint aggregation (Figure 4).

As the clearance decreases, the relative objective error decreases because the boundary of the union of exclusion spheres more closely approximates the true component boundary, reducing the conservative offset from the true feasible region. For the two-dimensional boundary patches of this problem ($n_{d_r} = 2$), halving the clearance requires four times as many discretization points. Without local aggregation, this directly quadruples the number of constraints and therefore the number of rows in the constraint Jacobian, causing the derivative evaluation time to scale quadratically with the inverse of the clearance. With local aggregation, the constraint count remains fixed, so the derivative evaluation time grows much more slowly, increasing only due to the larger number of SDF evaluations and aggregation operations per constraint.

B. Multi-cylinder Packing with Armadillo Obstacle

The multi-cylinder packing problem features one container cylinder, four package cylinders, and a fixed armadillo-shaped obstacle. Each package cylinder has all three translations and rotation about its two non-degenerate axes as design variables. The container cylinder radius and height are also design variables with each deformation defined relative to the container origin. The objective is to minimize the container volume, and geometric non-interference is enforced between every pair of distinct components, including the armadillo and the container cylinder. The problem is

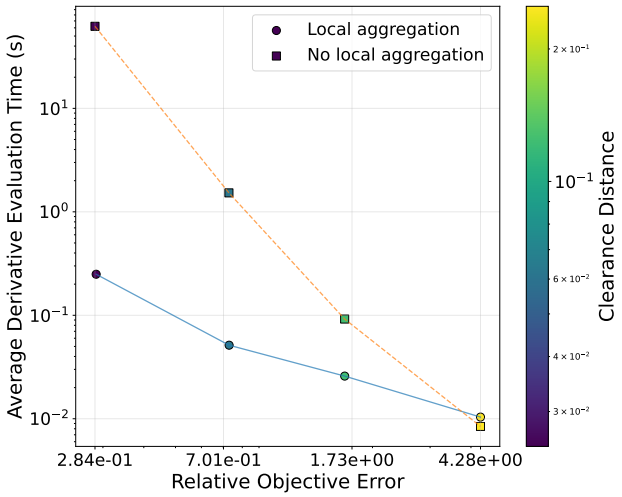


Fig. 4 The box packing problem is solved repeatedly with decreasing minimum clearance (color bar) both with and without local constraint aggregation. Decreasing the minimum clearance reduces the relative objective error as the exclusion sphere boundary more closely approximates the true component boundary. Without local aggregation (dashed orange, squares), the derivative evaluation time scales significantly with the inverse of the clearance because the constraint count grows with the discretization. With local aggregation (solid blue, circles), the constraint count remains fixed and the derivative evaluation time grows at a substantially lower rate, but still grows due to the increasing number of operations.

formulated as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & V(x) \\ \text{s.t.} \quad & c_g(x, \xi_I;\, j_a, j_b) \leq 0 \\ & \qquad \forall\, I \in \{1, \ldots, n_{I_{j_a}}\},\ \forall\, j_a \neq j_b \in \{1, \ldots, n_c\}, \end{aligned} \tag{21}$$

where V is the container cylinder volume and c_g is the geometric non-interference constraint function evaluated for each discretization point on each component boundary excluding the container cylinder, which is instead used only as the SDF target.

Using PySLSQP [31, 32] within modOpt [33], the optimization is solved using a formulation that uses SDFs but not the SDFs for the union of the components, a formulation mostly matching the proposed method with the component union SDFs but without local constraint aggregation, and a formulation using full aggregation (Figure 5). The formulation that didn't leverage union SDFs converged in 9 major iterations including 9 function evaluations. Since the number is equivalent, the line search took the full step at each operation, which can be a sign of good conditioning for the problem. The formulation using union SDFs, matching the proposed method, converged in 10 major iterations including 10 function evaluations when using an aggregation sharpness value of 80. Although the union SDF approach used an extra iteration, it still converged 8% faster due to a 16.64% decrease in the average model derivative evaluation time. The full aggregation approach converged in 91 major iterations including 192 model evaluations to a solution with 0.57% error in the optimized objective when using an aggregation sharpness value of 80. Although significantly more model evaluations were required, the full aggregation approach converged 92.6% faster for this problem compared to the union SDF formulation.

This increased speed when using full aggregation is likely because the cylinders were finely discretized, including areas that were likely inactive the whole optimization. As such, leveraging local constraint aggregation could lead to significant improvement in optimization time. Notably, since the model is purely geometric, decreasing the computational cost of the constraint Jacobian almost proportionally decreases the whole model derivative evaluation. In contrast, when solving MDO problems, the geometric non-interference constraint Jacobian computation time only makes up a small portion of the model derivative evaluation, decreasing the importance of the evaluation time of the non-interference constraint Jacobian and increasing the importance of decreasing the number of optimization iterations.

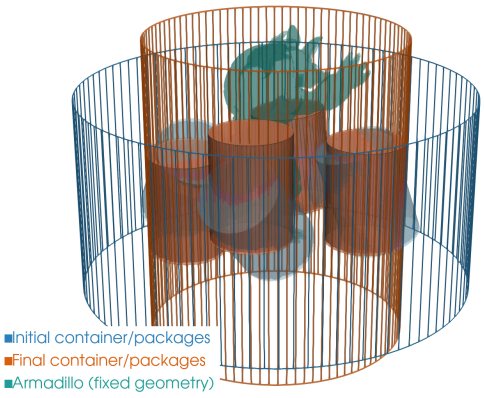


Fig. 5 The initial (blue) and optimized (orange) configurations are shown as a result of container volume minimization that must satisfy geometric non-interference between the container, package cylinders, and the armadillo.

C. Volume Minimization of a Blended-Wing-Body Aircraft with Payload

The BWB volume minimization problem uses a BWB geometry as the enclosing component. The payload consists of one oversized cargo box and 18 standard pallet boxes. The design variables are the center-body span and the four control points of a cubic B-spline parameterizing a chord-stretch function from center to tip. The objective is the BWB internal volume, approximated using a grid of points on the upper and lower surfaces. The full proposed method is applied. All payload components are enforced to be interior to the BWB, and each of the 18 pallets is enforced to be exterior to every other pallet. The oversized cargo payload is permitted to overlap with the pallets, as these represent mutually exclusive loading conditions. The problem is formulated as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & V(x) \\ \text{s.t.} \quad & c_g(x, \xi_I;\, j) + d_{\min} \leq 0 \\ & \qquad \forall\, I \in \{1, \ldots, n_{I_j}\},\ \forall\, j \in \{1, \ldots, n_c\}, \end{aligned} \tag{22}$$

with

$$c_g(x, \xi_I;\, j) := -\widetilde{d}_{s_{-j}}\left(p_j(x)(\xi_I);\, \rho\right), \tag{23}$$

where $d_{\min}$ is set to $\frac{1}{40}$ of the payload width. The number of discretization points is chosen to satisfy Eq. (17) with this value of $d_{\min}$. The spanwise direction requires a sufficiently dense discretization because the chord-stretch design variables vary spanwise. As such, spanwise gaps in the discretization would allow the optimizer to reduce the chord between discretization points, leading to large component boundary interpenetration.

Using PySLSQP [31, 32] within modOpt [33], the optimization converged in 22 major iterations with 22 function evaluations, indicating that the line search accepted the full step at every iteration. To benchmark against full aggregation, the problem was also solved over a sweep of aggregation sharpness values (Table 1). All tested sharpness values produced high objective error at convergence: because the KS overestimation grows as ρ decreases, low sharpness values push the optimizer to an overly conservative solution. At $\rho = 50$, the optimization failed to converge within 500 major iterations, with no meaningful progress in the final 350, indicating optimizer stagnation.

Unlike the box packing problem, the iteration count does not increase consistently with sharpness. This behavior is likely attributable to two factors. First, the problem is potentially already poorly conditioned, even for small aggregation sharpness values, so the optimizer relies heavily on the line search regardless of the aggregation formulation. Second, since the pallets are near their final configuration from the start, higher sharpness values place the initial guess closer to the feasible boundary of the aggregated constraint, reducing the number of iterations needed to satisfy it. The average relative derivative evaluation time a decreasing trend with sharpness, which is unexpected, likely reflects noise in the timing data. On average however, the relative derivative evaluation time is lower when using the full-aggregation approach. Neverttheless, the total optimization time is substantially higher for all full-aggregation cases, driven by the large increase in iteration count rather than per-iteration cost.

Table 1 BWB volume minimization benchmarked using the proposed method and full aggregation over a sweep of sharpness values ρ. Objective error, iteration counts, and normalized computation times are all reported relative to the proposed method. The proposed method (blue) shows significantly faster convergence and improved accuracy compared to the full-aggregation formulation.

	Obj. error (%)	Major iters	Func. evals	Rel. avg. deriv. time	Rel. opt. time
Proposed	—	22	22	1	1
$\rho = 5$	223	191	634	1.04	22.93
$\rho = 10$	105	223	618	0.88	22.09
$\rho = 20$	33	225	425	0.83	14.45

D. Large-Scale Multidisciplinary Design Optimization of a Blended-Wing-Body Aircraft

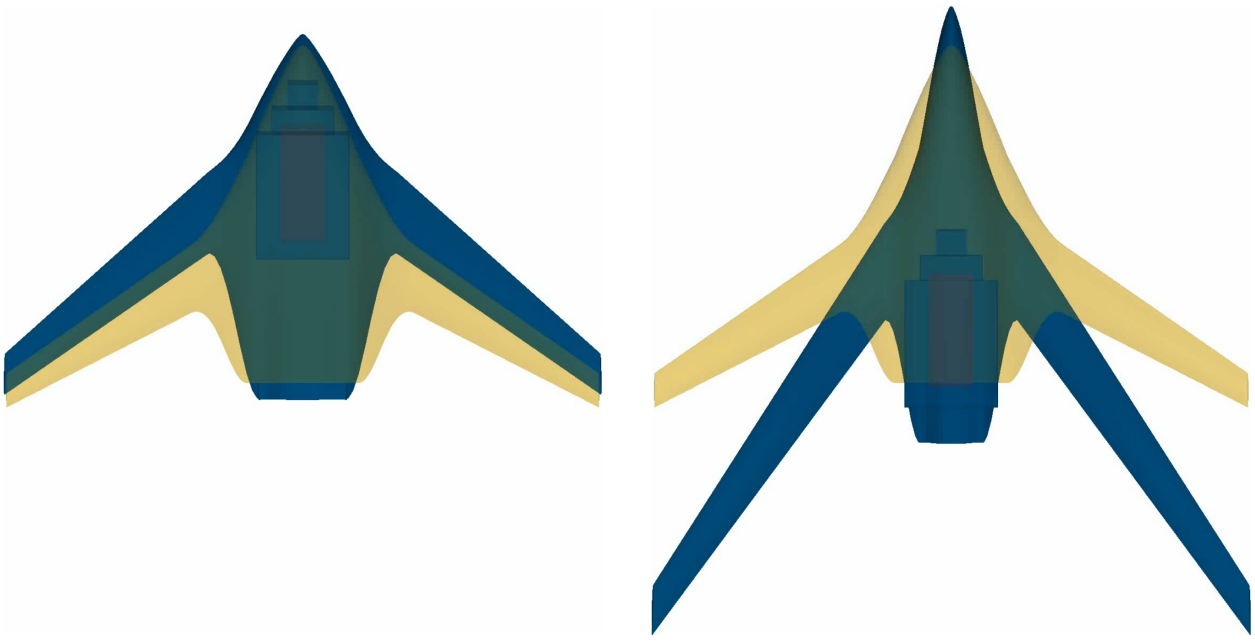
The large-scale MDO problem is an aero-structural optimization of the BWB aircraft minimizing fuel burn, adapted from Scotzniovsky et al. [34]. The design variables include structural thickness variables, cruise pitch, geometric twist and planform variables (including those from Section V.C), payload translations, and elevator deflection. As in the volume minimization problem, the payload consists of one oversized cargo box and 18 standard pallets as mutually exclusive payload cases. Since the BWB geometry strongly constrains the pallet layout and gradient-based optimizers have difficulty escaping local optima in the pallet arrangement, the 18 pallets parameterization are simplified to 3 standard pallets and 3 long pallets to reduce problem complexity. Furthermore, the parameterization is constructed to enforce that the 18 pallets move together.

The geometric non-interference constraints influence the design in two ways. First, they prevent the center-body from deforming to the extent that payload feasibility is violated for both payload cases. Second, because the payload has significant mass, the payload position affects the aircraft center of gravity, which in turn affects the static margin and trim moment constraints. Since there are two payload cases, the payload case with the forward-most center of gravity is used to compute the trim constraint, and the other payload is used to compute the static margin constraint. This formulation is conservative, but ensures that both payload cases can satisfy both constraints and does not require any additional aerodynamic model evaluations. The BWB geometry can therefore adapt to accommodate payload placement while satisfying these constraints.

Using PySLSQP [31, 32] within modOpt [33], the optimization converged in 162 major iterations and 340 function evaluations, with no component interpenetration in the converged solution (Figure 6a). The optimality and feasibility decreased relatively steadily throughout and converged, decreasing at least six orders of magnitude of their initial values (Figure 7a), suggesting the problem formulation is well-conditioned problem.

To assess the importance of geometric non-interference enforcement, the problem was also solved replacing the proposed constraints with a volume constraint requiring the internal BWB volume to be no less than its initial value (Figure 6b). Without geometric non-interference constraints, the volume constraint fails to prevent physically infeasible geometric configurations. The optimizer contracts the center-body spanwise until it can no longer accommodate the payload, while recovering the required volume through increased chord. It seems the absence of geometric constraints allows the payload to translate far aft, shifting the center of gravity rearward. For a BWB configuration without a tail and with a static margin lower bound of 10%, satisfying the trim and static margin constraints then favors increased wing sweep and washout rather than a more typical layout. In contrast, the proposed non-interference constraints force the payload to remain within the thick portion of the center-body, preventing this exploit and requiring the optimizer to satisfy trim primarily through elevator deflection, resulting in a physically realizable design. This comparison demonstrates that volume constraints are an inadequate surrogate for geometric non-interference when the optimizer has sufficient freedom to deform the enclosing geometry.

To benchmark the effect of the proposed partial aggregation approach, the problem is also solved using a fully-aggregated non-interference formulation in which the proposed constraints are replaced by a single KS-aggregated constraint with sharpness $\rho = 10$. The fully-aggregated formulation converged to a similar design but with a 0.96% higher objective value. The increase in the optimized objective is attributable to the conservative overestimation of the KS function, which pushes the optimizer away from the true feasible boundary. The optimality and feasibility histories of the fully-aggregated formulation are less steady than those of the proposed method, consistent with the ill-conditioning introduced by full aggregation. Compared to the fully-aggregated formulation, the proposed method



(a) Proposed Non-interference Constraints Formulation **(b) Volume Constraint**

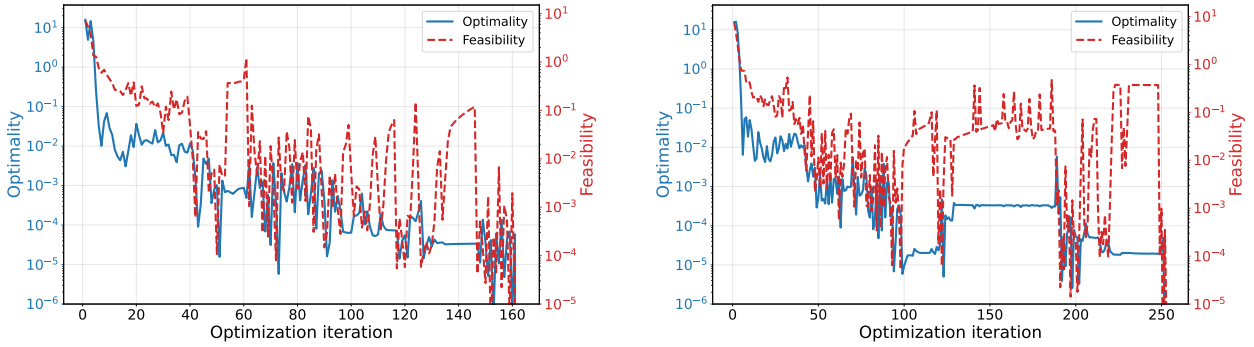
Fig. 6 Multidisciplinary design optimization is performed on a BWB aircraft with and without geometric non-interference constraints for two payload cases. The initial (yellow) and optimized (blue) geometries are plotted for optimizations with (left) and without (right) the geometric non-interference constraints. While the proposed non-interference constraints formulation method produces a physically realizable design, the formulation with just a volume constraint does not, permitting the payload to penetrate through the outer mold line and indirectly encouraging an atypical optimized design.

converged in 36% fewer major iterations and 67% fewer function evaluations, corresponding to a 46% reduction in total optimization time.

VI. Conclusion

This work proposed a method for enforcing geometric non-interference constraints efficiently and robustly in gradient based MDO. Starting from the continuous constraint formulation, the method applies partial aggregation in four steps: (1) an SDF is constructed for each component, aggregating the non-interference constraint across one component boundary per pair, (2) for each component, the other component SDFs are used to construct an SDF for the union of all other components, reducing the constraint count from $O(n_I \cdot n_c^2)$ to $O(n_I \cdot n_c)$, (3) a minimum clearance $d_{\min}$ is enforced at each discretization point to bound interpenetration between components, and (4) local constraint aggregation decouples the interpenetration bound from the number of optimization constraints. The method is designed to aggregate selectively, improving constraint scaling while limiting the effect of ill-conditioning introduced by aggregation methods.

The method was benchmarked on four problems of increasing complexity. On the box packing problem, the full aggregation method shows a decrease in objective error, but an increase required optimization iterations as the aggregation sharpness is increased. As such, the proposed method required 73% fewer major iterations and 77% less time than the full aggregation method to converge when the aggregation sharpness is large enough to converge the fully-aggregated problem with a relative objective error within 0.001. The multi-cylinder packing problem demonstrated enforcement of both interior and exterior non-interference relationships across multiple components, including a complex armadillo geometry. The BWB volume minimization problem showed that the method scales to a higher-dimensional design space with industry-relevant geometry and parameterization. Across a sweep of sharpness values, full aggregation required more iterations in all cases and produced higher objective error for small sharpness values due to constraint conservatism. The large-scale BWB MDO problem converged to a feasible design in 36% fewer major iterations and 46% less time compared to a formulation with fully-aggregated non-interference constraints, demonstrating efficient



(a) Proposed Non-interference Constraints Formulation **(b) Fully-aggregated Non-interference Constraint**

Fig. 7 The optimality (solid blue) and feasibility (dashed red) histories are plotted for each optimization iteration for the blended-wing-body multidisciplinary design optimization problem. The problem formulation facilitates a 6-order-of-magnitude decrease in both the optimality and feasibility over the course of the optimization. Compared to the fully-aggregated non-interference constraint formulation, the proposed method steadily decreases in optimality and feasibility and converges in fewer iterations.

enforcement on a realistic MDO problem.

Two limitations of the current method warrant future investigation. First, the constraint count scales as $O(n_I \cdot n_c)$, where n_I may be large for fine discretizations. For problems with very many components, the constraint Jacobian may become memory-prohibitive or slow to evaluate. More aggressive but still selective aggregation strategies could further reduce this scaling. Second, the spherical exclusion zones imposed by the minimum clearance condition do not conform to the component boundary shape, requiring a fine discretization to tightly approximate the true boundary. Future work could explore non-spherical exclusion zone formulations that more closely conform to the local boundary geometry, potentially reducing the required n_I.

By enabling robust and efficient enforcement of geometric non-interference constraints in gradient-based MDO, the proposed method broadens the class of problems to which MDO can be applied. More broadly, the partial aggregation framework introduced here, selectively aggregating to improve scaling while avoiding introducing ill-conditioning into regions of the problem that the optimizer is likely to explore, may be applicable to other classes of MDO constraints that exhibit similar structure.

Acknowledgments

The authors would like to acknowledge the Multidisciplinary Science and Technology Center of the Air Warfare Directorate, Air Force Research Laboratory for funding and supporting this effort through the Collaborative Center for Design and Research of Interdisciplinary Systems program. The authors would also like to thank Luca Scotzniovsky and Nicholas Orndorff for their help with the blended-wing-body optimization and Marius Ruh for supplying the projection-based signed distance function code implementation. AI was used to make minor edits to the text to improve readability, grammar, and language in accordance with AIAA's ethical standards.

References

[1] Zeidner, L. E., Turney, J. E., Mandel, R., Haydt, S. E., Lies, H., Behzadi, M., Zaffetti, P., Allison, J. T., Dunfield, N. M., Amato, N., et al., "Multi-Disciplinary Optimization for Packaging (MDOP)," 2024. URL `https://apps.dtic.mil/sti/html/trecms/AD1227289/index.html`.

[2] Dunn, R. C., Joshy, A. J., Lin, J.-T., Girerd, C., Morimoto, T. K., and Hwang, J. T., "Scalable enforcement of geometric non-interference constraints for gradient-based optimization," *Optimization and Engineering*, Vol. 25, No. 4, 2024, pp. 1849–1882. https://doi.org/10.1007/s11081-023-09864-2.

[3] LeDoux, S. T., Vassberg, J. C., Young, D. P., Fugal, S., Kamenetskiy, D., Huffman, W. P., Melvin, R. G., and Smith, M. F., "Study Based on the AIAA Aerodynamic Design Optimization Discussion Group Test Cases," *AIAA Journal*, Vol. 53, No. 7, 2015, pp. 1910–1935. https://doi.org/10.2514/1.J053535.

[4] Carrier, G., Destarac, D., Dumont, A., Méheut, M., Salah El Din, I., Peter, J., Khelil, S., Brezillon, J., and Pestana, M., "Gradient-Based Aerodynamic Optimization with the elsA Software," 2014. https://doi.org/10.2514/6.2014-0568.

[5] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. https://doi.org/10.2514/1.J053318.

[6] Birgin, E., Martınez, J., and Ronconi, D., "Optimizing the packing of cylinders into a rectangular container: A nonlinear approach," *European Journal of Operational Research*, Vol. 160, No. 1, 2005, pp. 19–33. https://doi.org/10.1016/j.ejor.2003.06.018, applications of Mathematical Programming Models.

[7] Brelje, B. J., Anibal, J. L., Yildirim, A., Mader, C. A., and Martins, J. R. R. A., "Flexible Formulation of Spatial Integration Constraints in Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2571–2580. https://doi.org/10.2514/1.J058366.

[8] Hajdik, H., Pacini, B., Yildirim, A., Brelje, B. J., and Martins, J. R., *Combined systems packaging and aerodynamic shape optimization of a full aircraft configuration*, 2023. https://doi.org/10.2514/6.2023-3589.

[9] Lin, J.-T., Girerd, C., Yan, J., Hwang, J. T., and Morimoto, T. K., "A Generalized Framework for Concentric Tube Robot Design Using Gradient-Based Optimization," *IEEE Transactions on Robotics*, Vol. 38, No. 6, 2022, pp. 3774–3791. https://doi.org/10.1109/TRO.2022.3180627.

[10] Kreisselmeier, G., and Steinhauser, R., "SYSTEMATIC CONTROL DESIGN BY OPTIMIZING A VECTOR PERFORMANCE INDEX," *Computer Aided Design of Control Systems*, edited by M. CUENOD, Pergamon, 1980, pp. 113–117. https://doi.org/10.1016/B978-0-08-024488-4.50022-X.

[11] Criado Risco, J., Valotta Rodrigues, R., Friis-Møller, M., Quick, J., Mølgaard Pedersen, M., and Réthoré, P.-E., "Gradient-based wind farm layout optimization with inclusion and exclusion zones," *Wind Energy Science*, Vol. 9, No. 3, 2024, pp. 585–600. https://doi.org/10.5194/wes-9-585-2024.

[12] Pentland, A., and Williams, J., "Good vibrations: modal dynamics for graphics and animation," *SIGGRAPH Comput. Graph.*, Vol. 23, No. 3, 1989, p. 207–214. https://doi.org/10.1145/74334.74355.

[13] Lutz, M., and Meurer, T., "Efficient Formulation of Collision Avoidance Constraints in optimization Based Trajectory Planning and Control," *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 228–233. https://doi.org/10.1109/CCTA48906.2021.9658663.

[14] Macklin, M., Erleben, K., Müller, M., Chentanez, N., Jeschke, S., and Corse, Z., "Local Optimization for Robust Signed Distance Field Collision," *Proc. ACM Comput. Graph. Interact. Tech.*, Vol. 3, No. 1, 2020. https://doi.org/10.1145/3384538.

[15] Chen, A. H., Hsu, J., Liu, Z., Macklin, M., Yang, Y., and Yuksel, C., "Offset Geometric Contact," *ACM Trans. Graph.*, Vol. 44, No. 4, 2025. https://doi.org/10.1145/3731205.

[16] Xu, H., Zhao, Y., and Barbi, J., "Implicit Multibody Penalty-Based Distributed Contact," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 20, 2014, pp. 1266–1279. URL https://api.semanticscholar.org/CorpusID:16553515.

[17] Kennedy, G. J., and Hicken, J. E., "Improved constraint-aggregation methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, 2015, pp. 332–354. https://doi.org/10.1016/j.cma.2015.02.017.

[18] Norato, J., Smith, H., Deaton, J., and Kolonay, R., "A maximum-rectifier-function approach to stress-constrained topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 65, 2022. https://doi.org/10.1007/s00158-022-03357-z.

[19] Barr, A. H., "Superquadrics and angle-preserving transformations," *IEEE Computer graphics and Applications*, Vol. 1, No. 01, 1981, pp. 11–23. https://doi.org/10.1109/MCG.1981.1673799.

[20] Osher, S., and Fedkiw, R., "Signed distance functions," *Level set methods and dynamic implicit surfaces*, Springer, 2003, pp. 17–22. https://doi.org/10.1007/b98879.

[21] Chen, J., and Ilieş, H. T., "Maximal Disjoint Ball Decompositions for shape modeling and analysis," *Computer-Aided Design*, Vol. 126, 2020, p. 102850. https://doi.org/10.1016/j.cad.2020.102850.

[22] Lutz, M., and Meurer, T., "Efficient Formulation of Collision Avoidance Constraints in optimization Based Trajectory Planning and Control," *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 228–233. https://doi.org/10.1109/CCTA48906.2021.9658663.

[23] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R., "Reconstruction and representation of 3D objects with radial basis functions," *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 2001, p. 67–76. https://doi.org/10.1145/383259.383266.

[24] Liu, S., Wang, C. C., Brunnett, G., and Wang, J., "A closed-form formulation of HRBF-based surface reconstruction by approximate solution," *Computer-Aided Design*, Vol. 78, 2016, p. 147–157. https://doi.org/10.1016/j.cad.2016.05.001.

[25] Drake, K. P., Fuselier, E. J., and Wright, G. B., "Implicit Surface Reconstruction With a Curl-Free Radial Basis Function Partition of Unity Method," *SIAM Journal on Scientific Computing*, Vol. 44, No. 5, 2022, p. A3018–A3040. https://doi.org/10.1137/22m1474485.

[26] Macêdo, I., Gois, J. P., and Velho, L., "Hermite Radial Basis Functions Implicits," *Computer Graphics Forum*, Vol. 30, No. 1, 2010, p. 27–42. https://doi.org/10.1111/j.1467-8659.2010.01785.x.

[27] Cuomo, S., Galletti, A., Giunta, G., and Marcellino, L., "Reconstruction of implicit curves and surfaces via RBF interpolation," *Applied Numerical Mathematics*, Vol. 116, 2017, p. 157–171. https://doi.org/10.1016/j.apnum.2016.10.016.

[28] Ruess, M., Schillinger, D., Bazilevs, Y., Varduhn, V., and Rank, E., "Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method," *International Journal for Numerical Methods in Engineering*, Vol. 95, No. 10, 2013, pp. 811–846. https://doi.org/10.1002/nme.4522.

[29] Chi, S.-W., Chen, J.-S., Hu, H.-Y., and Yang, J. P., "A gradient reproducing kernel collocation method for boundary value problems," *International Journal for Numerical Methods in Engineering*, Vol. 93, No. 13, 2013, pp. 1381–1402. https://doi.org/10.1002/nme.4432.

[30] Ventura, G., "An augmented Lagrangian approach to essential boundary conditions in meshless methods," *International Journal for Numerical Methods in Engineering*, Vol. 53, No. 4, 2002, pp. 825–842. https://doi.org/10.1002/nme.314.

[31] Kraft, D., "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988. URL https://degenerateconic.com/uploads/2018/03/DFVLR_FB_88_28.pdf.

[32] Joshy, A. J., and Hwang, J. T., "PySLSQP: A transparent Python package for the SLSQP optimization algorithm modernized with utilities for visualization and post-processing," *Journal of Open Source Software*, Vol. 9, No. 103, 2024, p. 7246. https://doi.org/10.21105/joss.07246.

[33] Joshy, A. J., and Hwang, J. T., "modOpt: A modular development environment and library for optimization algorithms," *Advances in Engineering Software*, Vol. 213, 2026, p. 104084. https://doi.org/10.1016/j.advengsoft.2025.104084.

[34] Scotzniovsky, L., Orndorff, N. C., Fletcher, A., Kao, J., and Hwang, J. T., "Demonstration of a Large-Scale Multidisciplinary Design Optimization Methodology on a Blended Wing Body Configuration using Mid-Fidelity, Multi-Point Analysis," *AIAA AVIATION FORUM*, 2026.