Highlights

Deflation constraints for global optimization of composite structures

Sankalp S. Bangera, Saullo G. P. Castro

- Propose a novel global optimization method based on deflation constraints
- Three types of deflation constraints are proposed: hypersphere, hypercube, hypercuboid
- Deflation constraints applicable to any optimizer supporting constraints
- Deflation constraints enable gradient-based optimizers to find global optima
- Successfully applied to the design of composite structures

Deflation constraints for global optimization of composite structures

Sankalp S. Bangera[a,1], Saullo G. P. Castro[a,*,1]

[a]Delft University of Technology, Aerospace Structures & Materials Kluyverweg 1, Delft, 2629 HS, The Netherlands

ARTICLE INFO

Keywords:
Deflation
Optimization
Constraints
Composites
Laminates
Variable Stiffness

ABSTRACT

The study presents deflation constraints that enable a systematic exploration of the design space during the design of composite structures. By incorporating the deflation constraints, gradient-based optimizers become able to find multiple local optima over the design space. The study presents the idea behind deflation using a simple sine function, where all roots within an interval can be systematically found. Next, the novel deflation constraints are presented: hypersphere, hypercube and hypercuboid; consisting of a combination of Gaussian and sigmoid functions. As a test case, the developed constraints are applied to the optimization of a double-cosine function, where all the 13 minima points could be found with 24 deflation constraints. It is shown that a new optimum is encountered after each deflation constraint is added, with the optimization subsequently re-started from the same initial point, or resumed from the last found minimum, being the latter the recommended approach. The new deflation constraints are then used in heuristic-based direct search methods, where a genetic algorithm optimizer is able to find new optimum individuals for straight-fiber composites. Lastly, variable-stiffness composites were designed with the deflation constraints applied to the multimodal optimization problem of recovering fiber orientations from a set of optimum lamination parameters.

1. Introduction

Structural optimization is a fairly vast subject in itself that is tightly associated with the design of lightweight structures. Composite materials have been a major enabler of these designs, owing to their superior specific material properties as compared to metals, and to their significantly larger design space enabled by their anisotropy. Both aeronautical and space industries have been continuously developing new concepts for composites to pursue these benefits, and Figure 1 shows the evolution of the use of composites in aeronautics over the last 60 decades. Note that Boeing's 787 Dreamliner was the first large commercial aircraft have 50% of the structural weight in composites (1).

1.1. Types of design space and optimization composites

Ghiasi et al. reviewed in detail different composite optimization methods used for constant (3) and variable (4) stiffness laminates, classifying the techniques into four categories, being in order of relevance: gradient-based, direct search and heuristic, specialized and hybrid methods. Gradient-based methods can find a local minimum generally faster than all the other methods, limited to contiguous problems with first or second derivatives (3). For problems that are large in dimension, with usually more than 30 variables (5), gradient-based methods may be the only viable optimization method (6). However, gradient-based optimizers can find the global optimum with high accuracy only in unimodal design spaces (7; 8), whereas in multimodal spaces the final solution largely depends on the initial point (3). A convex design space is a subset of a unimodal design space, as illustrated in Figure 2, and convexity requires that all line segments connecting any two points in the function are above the function and do not intersect it. Hence, not all unimodal functions are convex, but all multimodal functions are non-convex (6). As the design space becomes increasingly multimodal or non-convex, finding a global minimum becomes more difficult and would require special techniques that can evaluate multiple regions of the design space to find the point that best minimizes the function. These global search schemes involve different sampling or starting point approaches (9; 10) to increase the likelihood to converge to a global optimum, despite it is never guaranteed that a global optimum will be found, not even that the global search will converge onto a solution that is different from those previously found (6). In commercial software such as Altair's Optistruct, such global searches can be activated with the "dglobal" user input (11).

Direct search methods do not require derivatives, and can be more appropriately applied for composite lay-up design, handling a mixture of continuous and discrete variables. They can find the global optimum of multimodal objective functions, although with a significantly lower rate of convergence when compared to gradient-based methods (3). Genetic algorithms (GA) has been the most popular class of direct search method, according to Ghiasi et al. (3), with simulated annealing ranking second. Haftka and his group have pioneered the application of GA in composite design (12; 13; 14). António (15) proposed a hierarchical GA for multimodal optimization of hybrid composites with multiple solutions. The sunflower algorithm is another relevant global optimizer originally applied to inverse problem of structural damage detection in laminated composite plates, proposed by Gomes (16).

Variable stiffeness (VS) laminated composites can further increase the design space and hence the potential for better structural performance of composites (17), manufacturable by automated fiber placement (18) and continuous

*Corresponding author
S.G.P.Castro@tudelft.nl (S.G.P. Castro)
ORCID(s): 0000-0001-9711-0991 (S.G.P. Castro)
[1]The two authors had an equal contribution to this manuscript.

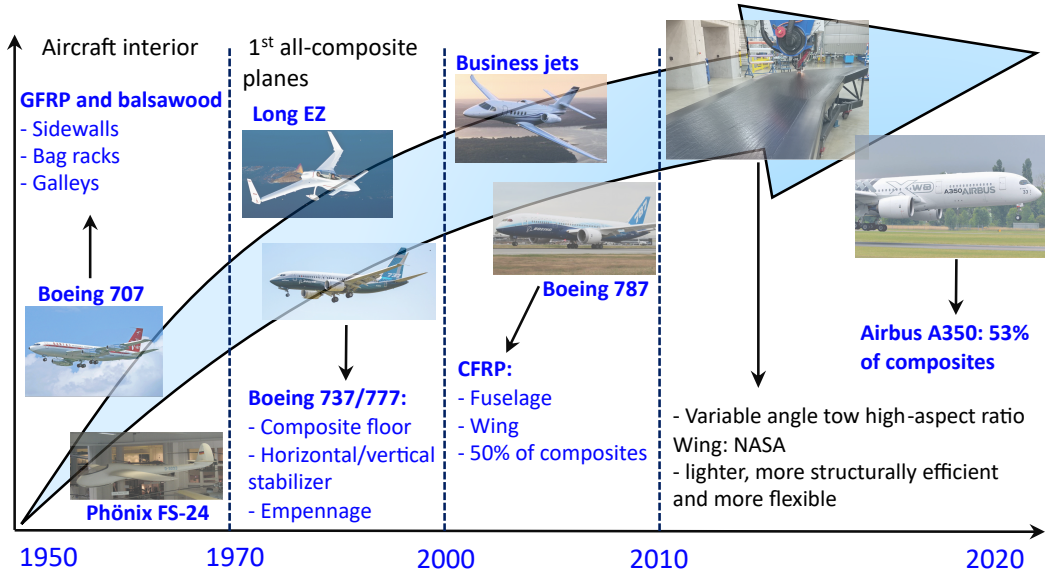


Figure 1: Utilization of fiber-reinforced composites in aircraft over the last decades, from Wang et al. (2).

two shearing (19). However, the larger design space created by VS laminates creates new challenges for the optimization, due to highly non-convex design spaces associated with the variable fiber angles (20).

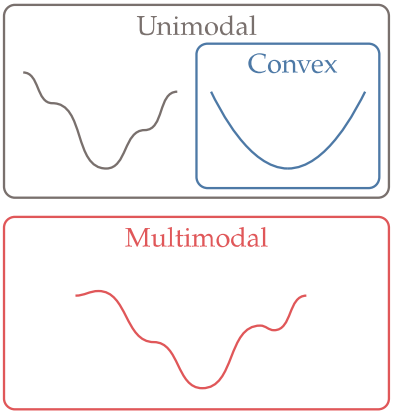


Figure 2: Unimodal and multimodal functions, from Martins & Ning (6).

Even in the design of constant stiffness laminated composites, the use of ply-angles as design variables is often associated with non-convexity (21). Miki (22) proposed the use of lamination parameters, introduced by Tsai & Pagano in 1968 (23), as design variables to render a convex optimization. Scardaoni & Montemurro (24) proved non-convexity of the feasible domain for both anisotropic and orthotropic-membrane laminates, even when those are parameterized using lamination parameters. However, approximations that lead to convex feasible domains have been proposed, such as the one for the in-plane and flexural lamination parameters developed by Fukunaga & Sekine (25; 26). Even when the optimization of the lamination parameters is performed in a convex design space, the next step of retrieving the fiber angles involves a multimodal optimization that is ubiquitously done using GA algorithms. In the present study, it is shown that this retrieval can be done with gradient-based optimizers when deflation constraints are used.

1.2. The deflation method

1.2.1. Deflation of scalar-valued functions

The first instance of the deflation methods were applied for systematic root finding in nonlinear functions (27), providing an easy visualization of the concept. Assume $p(x)$ to be a scalar-valued nonlinear function having n multiple roots $x_1, x_2, ..., x_n$, with each root found using an iterative method such as the Newton-Raphson (28). After having evaluated the initial root of $p(x)$, more roots can be systematically found by considering the following deflated function (29):

$$q(x) = \frac{p(x)}{\prod_{i=1}^{n}(x - x_i)}, \tag{1}$$

for which the already obtained roots can be effectively removed by the multiplicative term in the denominator. Consider the sine function $p(x) = sin(\pi x)$ with 7 roots within $-\pi \leq x \leq +\pi$. Figure 3 depicts $p(x)$ and the two new functions $q_1(x)$ and $q_2(x)$, created after two consecutive deflation steps, respectively after finding the roots at $x = -2$ and $x = +1$, given as:

$$\begin{aligned} q_1(x) &= \frac{\sin(\pi x)}{x - (-2)} \\ q_2(x) &= \frac{\sin(\pi x)}{(x - (-2))(x - (+1))} \end{aligned} \tag{2}$$

Figure 3 shows that the deflated functions have the specific roots completely removed, i.e. the deflated functions no longer cross zero at those points, while keeping all the other roots unchanged. This property of the deflation technique can be exploited to remove selected minima points from an arbitrary objective function, as demonstrated later in the present study.

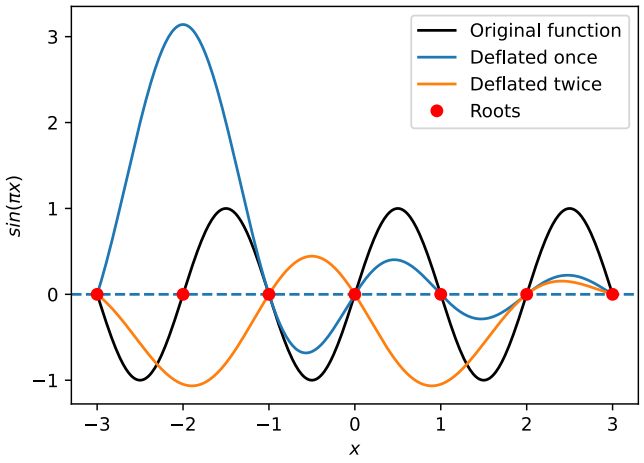


Figure 3: Deflated sine function obtained using Equation 1

1.2.2. Deflation applied to nonlinear equations

The idea behind polynomial deflation was generalized to partial differential equations (PDE) by Farrell et al. (29), extending the concept of deflation matrix M and deflation operator m that had been previously introduced by Brown & Gearhart (30), also with the intent of solving PDEs. Farrell proved that convergence to different solutions from the same starting point was achieved after deflation. Consider the residuals for a system of n nonlinear equations $F(x)$ such that:

$$F(x) = 0 \tag{3}$$

After solving the system of equations once, solution x_1 is obtained, such that a deflation matrix can be calculated using a deflation operator:

$$\begin{aligned} M(x; x_1) &= m(x; x_1)\mathcal{I} \\ m(x; x_1) &= ||x - x_1||^{-p} + \sigma \end{aligned} \tag{4}$$

where $\mathcal{I}$ is the identity matrix ($n \times n$); p is the pole strength or power that dictates the rate at which the function approaches infinity; and σ is an offset parameter that is reached when the norm distance $||x - x_1|| \to \infty$ (31). The deflated nonlinear system of equations $G(x)$ becomes (31):

$$G(x) \equiv M(x; x_1)F(x) = 0 \tag{5}$$

The system in Equation 5 can be solved to obtain a new solution x_2. Here G satisfies the following properties (31):

1. The two system of equations, $F(x) = 0$ and $G(x) = 0$ both have the same solutions for all $x \neq x_1$.
2. With the known solution x_1, G will not converge again to x_1 under the assumption $\lim_{x \to x_1} \|G(x)\| > 0$

After having found $\tilde{K}$ multiple solutions, the method in (29) proposes multiplying the deflation matrices and solving the following set of equations:

$$\prod_{k=1}^{\tilde{K}} M(x; x_k)F(x) = 0. \tag{6}$$

Farrell et al. in a recent study (32) have further extended this method for semi-smooth equations. Deflation does not guarantee that all the solutions to a problem are found; however, it provides a systematic method to explore a progressively larger number of nonlinear solutions.

1.2.3. Deflation in global search

In the context of sizing optimization, the idea to apply deflation as a new constraint has been independently developed by the authors, with the only literature reference using a similar approach for topology optimization being the non-peer reviewed pre-print by Tarek & Huang (33). Consider the nonlinear programming (NLP) problem for which multiple solutions need to be evaluated:

$$\begin{aligned} \underset{x \in \mathcal{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) = 0 \\ & l \leq x \leq u \end{aligned} \tag{7}$$

where l and u are respectively the lower and upper bounds of the design variable x. This constrained minimization problem can be converted into an unconstrained problem by expressing the objective function as the Lagrangian, expressed as (33):

$$\mathcal{L}(x, \lambda, z_+, z_-) = f(x) + c(x)^T \lambda + (x - u)^T z_+ - (x - l)^T z_- \tag{8}$$

where $\lambda \in \mathcal{R}^m$ is the Lagrange multiplier vector associated with the equality constraints, $z_-, z_+ \in \mathcal{R}^n{}_+$ is the Lagrange multiplier vectors associated with the lower and upper bound. The Karush-Kuhn-Tucker (KKT) conditions are satisfied when (33):

$$\begin{aligned} \nabla_x \mathcal{L}(x, \lambda, z_+, z_-) &= 0 \\ c(x) &= 0 \\ l \leq x &\leq u \\ z_+ &\geq 0 \\ z_- &\geq 0 \\ (x - u)^T z_+ &= 0 \\ (x - l)^T z_- &= 0 \end{aligned} \tag{9}$$

A point x that satisfies these conditions is therefore known as a KKT point, and finding multiple solutions to

the NLP consists of finding multiple KKT points. In the work of Tarek & Huang (33) two versions of the deflation constraint were proposed. First, an additional variable y is introduced into the optimization problem along with the deflation constraint m. With $y \geq 0$, the deflation constraint is given as (33):

$$m(x; x_1) = ||x - x_1||^{-p} + \sigma \leq y \tag{10}$$

where x_1 is the first optimum point that is found previously, p is a power (usually varies between 2-4), and an offset term σ. After this constraint becomes active, it is considered that if the new found optimum point x_2 is associated with variable y such that y is finite and has a value in exact arithmetic, then $x_1 \neq x_2$ (since $\lim_{\boldsymbol{x}\to\boldsymbol{x}_1} m(x; x_1) = \infty$). It can further be proved that every KKT point of the new NLP with a finite value of y is a KKT point of the original formulation (33). A proof of how (x^*, y^*) would be a regular KKT point for the modified problem after the introduction of the deflation constraint is provided in Appendix A.

If it is assumed that $m > 0$ and $y > \sigma$, the deflation constraint would be equivalent to the distance constraint as follows (33):

$$||x - x_1||^p \geq z \tag{11}$$

where $z = 1/(y - \sigma)$. Thus, it becomes clear that the deflation constraint essentially puts a constraint on the known solutions. If at any point, the optimizer approaches $z = 0$ or $y = \infty$, then the deflation operator is not distancing the new points from the known optimum points sufficiently. Hence, in this case, the main hyperparameters that can be varied to ensure that this distancing occurs are the offset σ and the power p, as shown in Equation 10.

The first deflation constraint defined in Equation 10 can be modified by omitting the additional variable y, and replacing it with a large finite constant $\boldsymbol{M}$, such that a second deflation constraint is obtained as (33):

$$m(x; x_1) = ||x - x_1||^{-p} + \sigma \leq \boldsymbol{M} \tag{12}$$

Because there are no additional variables to this second deflation constraint, a proof similar to what is given in Appendix A can be used to conclude that the newly found point x^* is a regular KKT point to Equation 7. So far, Equation 10 and Equation 12 have been evaluated with a single known solution, whereas for $\tilde{K}$ known solutions the deflation constraint can be expressed as a summation, given by (33):

$$\sum_{k=1}^{\tilde{K}} m(x; x_k) \leq y \tag{13}$$

1.3. Contributions of the present work

Given the challenges related to performing systematic global search in multimodal design spaces, the present study proposes a novel deflation constraint that is included in the optimization problem by minimizing the Lagrangian instead of the main objective function, thus being compatible with any algorithm that allows constrained optimization.

The methodology of Tarek & Huang (33), similarly to the methodology initially attempted by Bangera (34), does not perform adequately in a larger design space after the first few deflation constraints are added. This lack of robustness is mainly due to their deflation constraint not being differentiable at the deflated points, and due to the relatively arbitrary offset parameter σ. The deflation constraint herein proposed addresses these shortcomings, enabling a robust exploration of larger design spaces by guaranteeing the discovery of new distinct minima when the optimization is re-started or resumed, after a new deflation constraint is added. This new constraint is applied without altering the optimization algorithm, making it suitable to both gradient-based and gradient-free optimization methods.

2. Methodology

2.1. Interior-point optimization algorithm

Interior-point algorithms stems from interior penalty methods that associate a penalization term with the constraints, being a key difference the fact that the constraints are not directly penalised. In interior-point, the penalization acts on slack variables that are added to the constraints, with the penalty term increasing as the optimizer moves towards the boundary of the constrained domain. The formulation is written as (6):

$$\begin{aligned} \underset{x,s}{\text{minimize}} \quad & f(x) - \mu_b \textstyle\sum_{j=1}^{n_g} \ln s_j \\ \text{subject to} \quad & h(x) = 0 \\ & g(x) + s = 0 \end{aligned} \tag{14}$$

where μ_b is a barrier parameter. The resulting formulation turns the inequality constraint into an equality constraint with the addition of the slack variables s.

Newton's method can be applied to solve the KKT system of equations in Equation 14, where the logarithm term associated with the slack variables is only defined for positive s values, acting as a barrier for negative s values (see Figure 4). Owing to the positive s values, $g(x^*) < 0$ at the solution, hence satisfying the inequality constraints (6).

The interior-point formulation of Equation 14 is equivalent to the original constrained problem when $\mu_b \to 0$. Hence, a sequence of solutions needs to be obtained such that $\mu_b \to 0$. A constrained problem can be reformulated as an unconstrained problem by utilizing the Lagrangian function along with associated Lagrange variables, or Lagrange multipliers (6). The Lagrangian for the interior-point optimization problem can be written as (6):

$$\mathcal{L}(x, \lambda, \sigma, s) = f(x) + \mu_b e^\top \ln s + h(x)^\top \lambda + (g(x) + s)^\top \sigma, \tag{15}$$

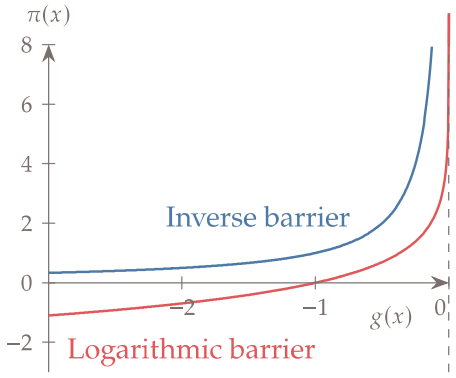


Figure 4: Plot of inverse barrier and Logarithmic barrier penalty functions (6)

where, $\ln s$ is a vector having n_g components being the logarithms of each value of s; $e = [1, ..., 1]$ is a vector of length n_g containing ones, introduced to express the sum in vector form; λ is the Lagrange variable vector associated with the equality constraints; and σ is the Lagrange variable associated with the inequality constraints. The KKT conditions can be derived by taking the derivatives with respect to x, λ, σ and s, leading to:

$$\begin{aligned} &\nabla f(x) + J_h(x)^\top \lambda + J_g(x)^\top \sigma = 0 \\ &h = 0 \\ &g + s = 0 \\ &-\mu_b e + S\sigma = 0 \end{aligned} \tag{16}$$

where S is a diagonal matrix with its diagonal values given by the slack variable vector. With this set of residual equations, Newton's method can be applied. Taking the Jacobian of the equations in Equation 16, the following linear system is obtained:

$$\begin{bmatrix} H_L(x) & J_h(x)^\top & J_g(x)^\top & 0 \\ J_h(x) & 0 & 0 & 0 \\ J_g(x) & 0 & 0 & \mathcal{I} \\ 0 & 0 & S & \Sigma \end{bmatrix} \begin{bmatrix} p_x \\ p_\lambda \\ p_\sigma \\ p_s \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(x, \lambda, \sigma) \\ h(x) \\ g(x) + s \\ S\sigma - \mu_b e \end{bmatrix} \tag{17}$$

where Σ is a diagonal matrix whose entries are given by the values of vector σ; and $\mathcal{I}$ is the identity matrix. For numerical efficiency, the system can be made symmetric after multiplying the last rows in Equation 17 by S^{-1}, which can be calculated as $S^{-1}_{kk} = 1/s_k$. This results in the following symmetric linear system that can be directly solved:

$$\begin{bmatrix} H_L(x) & J_h(x)^\top & J_g(x)^\top & 0 \\ J_h(x) & 0 & 0 & 0 \\ J_g(x) & 0 & 0 & \mathcal{I} \\ 0 & 0 & \mathcal{I} & S^{-1}\Sigma \end{bmatrix} \begin{bmatrix} p_x \\ p_\lambda \\ p_\sigma \\ p_s \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(x, \lambda, \sigma) \\ h(x) \\ g(x) + s \\ \sigma - \mu_b S^{-1} e \end{bmatrix} \tag{18}$$

The optimization algorithm developed in the present study fulfils the requirement of handling versatile inputs, being applicable to constrained and unconstrained optimization problems, while handling both equality and inequality constraints. Therefore, generally grouped in four types of optimization problems:

1. Fully unconstrained, which can be solved by applying Newton's method to the simple unconstrained conditions of optimality.
2. Only with equality constraints that can be solved by employing the method of sequential quadratic programming, which involves a similar set of equations as the interior-point method (Equation 18), but omits the equations associated with the inequality constraints.
3. Only with inequality constraints, which can be solved using the interior-point method (Equation 18), but omitting the equations associated with the equality constraints.
4. With both equality and inequality constraints, which can be solved using the interior-point method.

2.2. Deflation constraint

Consider the deflation constraint of Equation 4 evaluated at the design point $x = x^*$:

$$m(x; x^*) = ||x - x^*||^{-p} + \sigma \tag{19}$$

Figure 5 plots Equation 19 using $x^* = 0$ and $p = 2$. Here, the offset parameter σ influences the ability to overcome saddle points created by the deflation constraint when using gradient-based optimizers, but the use of σ keeps the influence of the deflation constraint even far from $x = x^*$. When $||x - x^*|| \to 0$, the constraint behaves as $m(x; x^*) \to \infty$, which is a discontinuity of the deflation constraint that prevents its robust use in gradient-based optimizers.

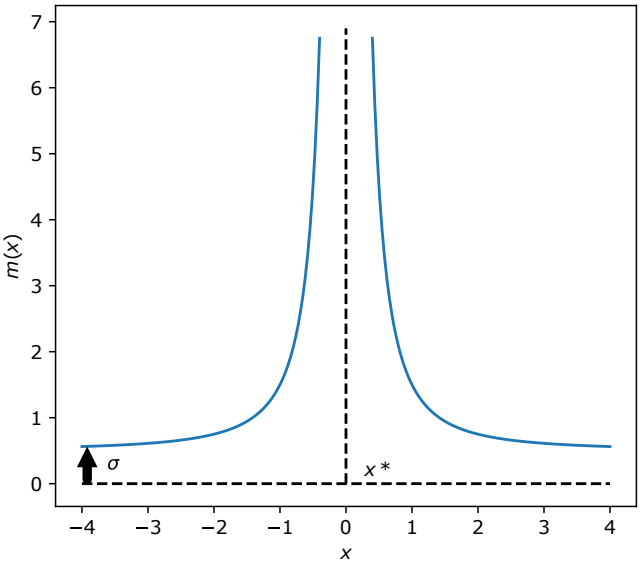
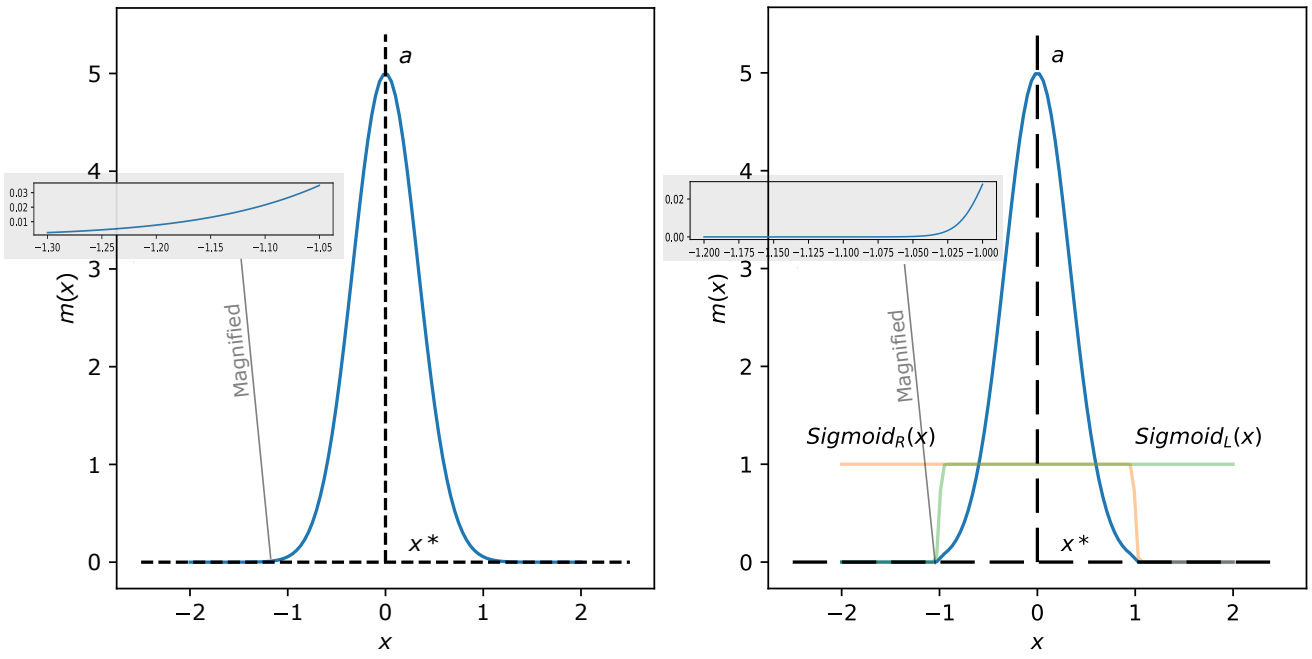


Figure 5: Plot of deflation function given by Equation 19, also proposed by Tarek & Huang (33)

Aiming to reach a differentiable deflation constraint function that has no influence outside the proximity of $x = x^*$, the present study proposes the use of the following



(a) Gaussian deflation function (Equation 20)

(b) Gaussian deflation function with the sigmoid terms (Equation 23)

Figure 6: Comparison of the proposed Gaussian deflation with and without the sigmoid terms.

Gaussian function as one of the main components of the new deflation function:

$$m(x; x^*) = ae^{-\frac{(x-x^*)^2}{2c^2}} \tag{20}$$

where a is the maximum amplitude; and c is the standard deviation. It is proposed to utilise the total span of the desired deflation divided by 6 as the standard deviation, such that 99.7% of the Gaussian distribution is covered.

Figure 6a depicts the Gaussian distribution using $a = 5$ and $c = 1/3$. For both the original deflation function, depicted in Figure 5, and the proposed Gaussian deflation function depicted in Figure 6a, it can be observed that the deflation constraint does not terminate to zero beyond the deflation span $-1 \leq (x - x^*) \leq +1$, nor it becomes negative, making it inappropriate to be used as a new equality or inequality constraint. This is because the deflation constraint function still has a residual value $m(x; x^*) > 0$ that extends way beyond the desired deflation span. Therefore, it is suggested to use a differentiable decay function, being a sigmoid, that multiplies to the Gaussian function with the aim to terminate the aforementioned residual value entirely beyond the desired deflation span. Two one-sided sigmoid functions are used, given as (35):

$$Sigmoid_L(x) = \frac{1}{1 + e^{-K(x-x^*+b)}} \tag{21}$$

$$Sigmoid_R(x) = \frac{1}{1 + e^{K(x-x^*-b)}} \tag{22}$$

where b is half of the required deflation span; K is a parameter that controls the steepness of how the sigmoid function goes from 1 to 0 at the required point of termination, i.e. at $x - x^* = b$. With the two one-sided sigmoid functions, Equation 21 and Equation 22, it is possible to make $m(x; x^*) \to 0$ respectively for negatives and positives values of $||x - x^*||$; with the $-b$ term becoming $+b$ and K becoming $-K$, from Equation 22 to Equation 21. Hence, within the desired deflation span, the product of the two sigmoid functions results in 1, whereas outside the span this product results in 0.

Figure 6b shows how the Gaussian function is changed by the multiplying sigmoid terms, resulting in the following deflation function:

$$m(x; x^*) = ae^{-\frac{(x-x^*)^2}{2c^2}} \cdot Sigmoid_L(x) \cdot Sigmoid_R(x) \tag{23}$$

where $c = b/3$. In Figure 6b, $b = 1$ and $K = 100$ and the resulting deflation function terminates to zero beyond $||x - x^*|| = 1$. Noticeably, the deflation function value does not become zero immediately after crossing 1, but this can be adjusted using a higher value of K for a steeper termination, or a parameter b that is slightly smaller than half of the required deflation span. Note that, the associated Gaussian and sigmoid functions increase with the number of dimensions being deflated in an optimization problem.

Three deflation constraint methods are herein proposed: hypersphere, hypercube and hypercuboid; being the last two based on Equation 23, with their advantages and drawbacks discussed in the following.

2.2.1. Hypersphere approach

The hypersphere or n-sphere approach is a distance-based approach in which the deflation span region is defined by a hypersphere. The main advantage is that only one scalar parameter controls the size of the deflated region, i.e. the radius of action r that is the same over all dimensions undergoing deflation within the optimization problem.

The distance constraint is given as:

$$||x - x^*|| \leq r \tag{24}$$

which whether the current design point x falls within the required deflation span. Followed by that, the deflation constraint of Equation 20 can be directly applied, without requiring the sigmoid decay function. Deflation is not applied when $||x - x^*|| > r$.

2.2.2. Hypercube approach

Here, a hypercube is used to represent the required deflated span, where each side of the hypercube has the same length represented by the scalar r. The distance constraint is given as:

$$x_i - x_i^* \leq r \tag{25}$$

where $i = 1, 2, \ldots, n$; with n being the number of dimensions of the problem. For an $n-$dimensional problem, the deflation function becomes:

$$m(x; x^*) = \prod_{i=1}^{n} \left(a e^{-\frac{(x_i - x_i^*)^2}{2c^2}} \cdot Sigmoid_L(x_i) \cdot Sigmoid_R(x_i) \right) \tag{26}$$

Figure 7a plots the hypercube deflation function of Equation 26 for a 2-dimensional problem, showing the even spread across all dimensions, here using a value of $r = 4.5$.

2.2.3. Hypercuboid approach

When the deflated region is represented by a hypercuboid, a different length is attributed to each side of the cuboid, being the number of sides equal to the number of dimensions being deflated within the optimization problem. The $n-$dimensional vector of deflation spans is $r = \{r_1, r_2, \ldots, r_i, \ldots, r_n\}$; such that the distance constraint becomes:

$$x_i - x_i^* \leq r_i \tag{27}$$

where $i = 1, 2, \ldots, n$. The expression for the deflation function becomes:

$$m(x; x^*) = \prod_{i=1}^{n} \left(a e^{-\frac{(x_i - x_i^*)^2}{2c_i^2}} \cdot \frac{1}{1 + e^{K(x_i - x_i^* - r_i)}} \cdot \frac{1}{1 + e^{-K(x_i - x_i^* + r_i)}} \right) \tag{28}$$

with $c_i = r_i/3$ for $i = 1, 2, \ldots, n$, being compatible with the deflation span vector r. Figure 7b illustrates Equation 28 for a $2-$dimensional problem, where deflation constraint is uneven across different dimensions, here using $r = \{2, 4.5\}$.

2.2.4. Advantages and drawbacks

The idea behind having a deflation constraint is to have a simple approach that can be applied to any optimization method intuitively seamlessly to find different solutions in a multimodal design space. The three proposed deflation constraints can be applied to any optimization problem; however, there are certain cases in which some perform better than the others.

The hypersphere and hypercube approach have only one distance parameter in the deflation problem, i.e. the scalar r, being therefore adequate when all design variables should be equally deflated from existing minima points. Even though it is simpler to have a single parameter to control the size of the deflated region, it increases the chances of masking local minima throughout the design space. This scenario is shown in Figure 8, which could happen in any of the three proposed deflation constraints. The figure displays two local minima in green that were masked due to r being too large. Moreover, Figure 8 illustrates that at the point of termination of the deflation constraint, the objective function intersects the deflated region creating false-minima depicted in red. These false-minima become possible minima to be discovered by the optimizer in subsequent searches after the deflation constraint is applied, and can be overcome when further deflation constraints are added. Masking of minima points should be always avoided, which is possible by using smaller values of r, with the side effect of increasing the number of deflations that are necessary to move away from a local minimum region to the next.

However, when it is desired that different design variables are deflated differently, the hypercuboid approach provides flexibility in selecting the required r_i values along each dimension.

3. Results and discussions

The proposed deflation constraints are applied across various problems to evaluate the capabilities of the method. First, a test case study based on a double-cosine function is presented. The next two case studies represent different sizing problems encountered in the design of composites.

3.1. Test case: double-cosine function

The double-cosine function of Equation 29 is used to verify whether the proposed deflation constraint can be used to obtain all the 13 minima points within a bounded region of the objective function.

$$f(x_1, x_2) = -\cos(x_1 \pi)\cos(x_2 \pi) \tag{29}$$

The results below were generated for the three deflation constraint approaches herein proposed: hypersphere,

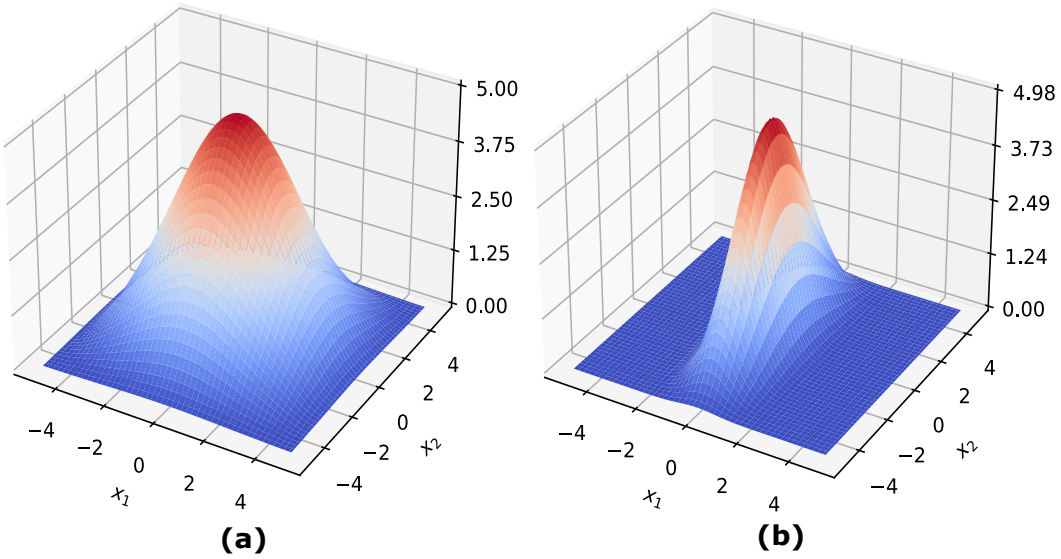


Figure 7: **(a)** Plot of Gaussian deflation function distribution using hypecube approach (Equation 25) **(b)** Plot of Gaussian deflation function distribution using hypercuboid approach (Equation 28)

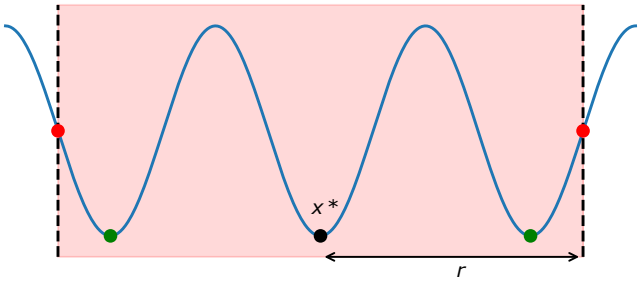


Figure 8: Masking of local minima (green) due to a large value of r, and creation of false minima (red) at the boundaries of the deflated region.

hypercube and hypercuboid. Here, the same starting point for the optimizer is used for all cases, even after deflation is performed. However, as shown in Appendix B, it was observed with the double-cosine function that consistently fewer deflation iterations are necessary if the starting point after deflation is be set to the latest found solution, such that the search for a new solution proceeds from there.

3.1.1. Hypersphere

The hypersphere deflation constraint is applied to the double-cosine function within the bounded region $-2.5\pi \leq x_1, x_2 \leq +2.5\pi$ and using as starting point $x_1, x_2 = -5.5, -5.5$. The radius of the deflation hypersphere is $r = \pi$. Figure 9 shows a plot of the double-cosine function, where the dashed regions represent negative values; the red dot depicts the starting point of the optimizer; whereas the black dots are the obtained solutions. After deflating the function 29 times, all the 13 minima within the bound space have been discovered. Note the false minima points are created at the end of the regions intersecting with the deflation radius, represented as red dots in Figure 8. The presence of false minima is not a problem for the overall optimization result given that

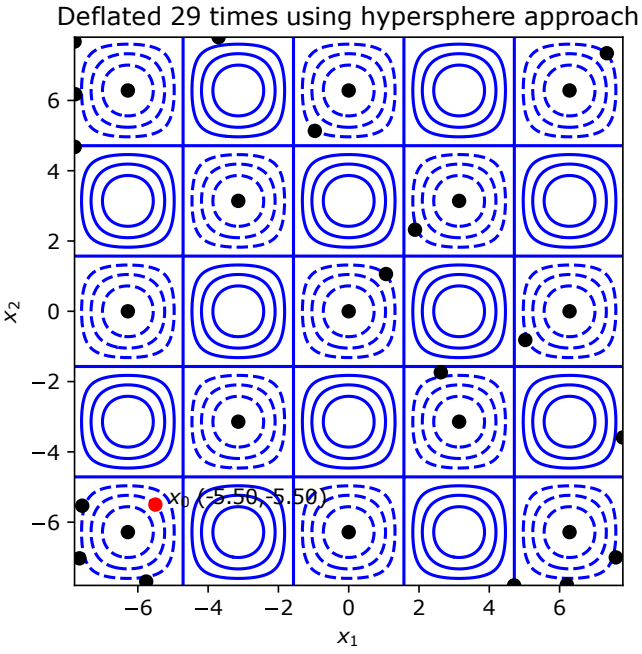


Figure 9: Hypersphere approach applied to locate minima of the double-cosine function (Equation 29)

enough additional deflations are carried on, because when new deflation constraints are added the false minima points are eventually overcome. The value of deflation radius $r = \pi$ has shown to be small enough, preventing local minima of being masked, e.g. the green dots in Figure 8. Hence, to guarantee the robustness of the method, it is recommended to use small values for the deflation radius r, despite a possible increase in the number of deflation iterations needed to discover the required solutions.

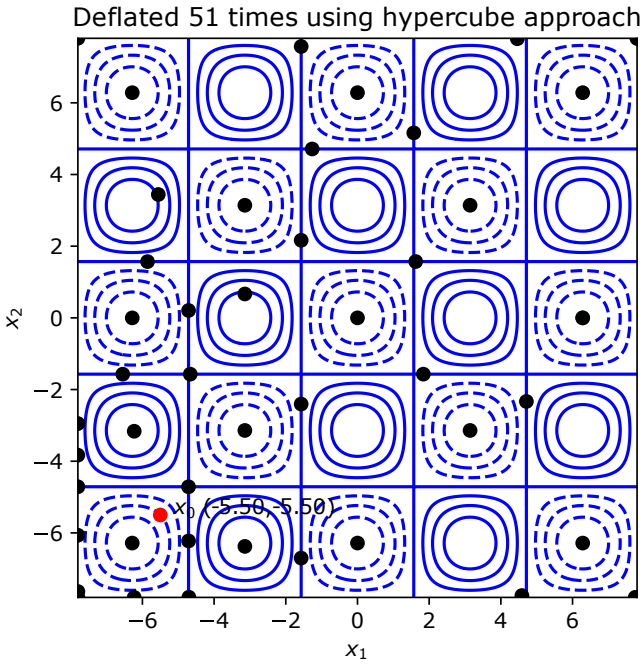


Figure 10: Hypercube approach applied to locate minima of the double-cosine function (Equation 29)

3.1.2. Hypercube

For the hypercube, the bounds and starting point are the same as for the hypersphere, with the side of the hypercube set to $r = pi$. As seen in Figure 10, 51 deflation iterations are required to find all 13 minima points, as compared to 29 of the hypersphere. However, this difference is not verified when the starting point for the new optimization, after each deflation step, is set to be the latest minimum point (see Appendix B), with both the hypersphere and the hypercube approaches requiring 24 deflation iterations to find all the 13 minima points.

3.1.3. Hypercuboid

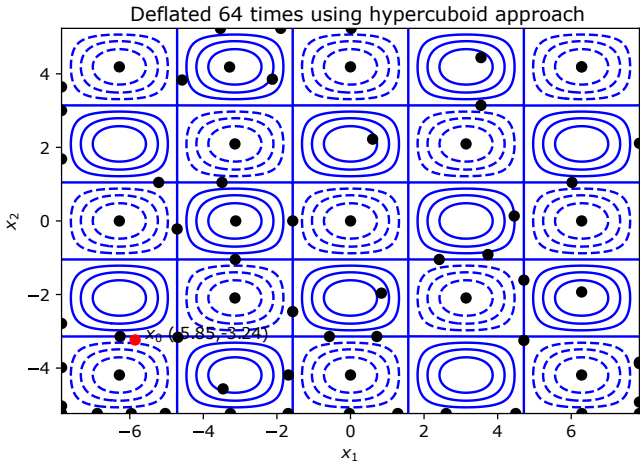


Figure 11: Hypercuboid approach applied to locate minima of the double-cosine function (Equation 29)

For the hypercuboid, the double-cosine function is skewed in one dimension, as follows:

$$f(x_1, x_2) = -\cos(x_1\pi)\cos(2x_2\pi) \tag{30}$$

where it can be seen that each sub-space in the whole design space is now shaped like a rectangle, making a proper test for the hypercuboid approach. The design space is bounded such that $-2.5\pi \leq x_1 \leq +2.5\pi$, and $-1.5\pi \leq x_2 \leq +1.5\pi$. The dimensions used for the 2-sided deflation hypercuboid (i.e. a rectangle in two-dimensions) are, $r_1 = \pi$, along x_1; and $r_2 = \pi/2$, along x_2. As seen in Figure 11, the function has to be deflated 64 times to yield all the 13 minima within the bounded region. When the starting point after each deflation is set to be the latest minimum point, as shown in Appendix B, the hypercuboid approach required only 44 deflations to find all the 13 minima.

3.2. Application to composite design

Case study 1 presents a genetic algorithm-based optimization seeking minimum mass and constrained by buckling and strength, where the laminate is parameterized using discrete ply orientations. Case study 2 presents a gradient-based optimization, where the laminate is parameterized using lamination parameters (LP) and the total thickness, typically applied in the design of variable-stiffness (VS) laminates.

3.2.1. Case study 1: deflation in gradient-free discrete-based optimization

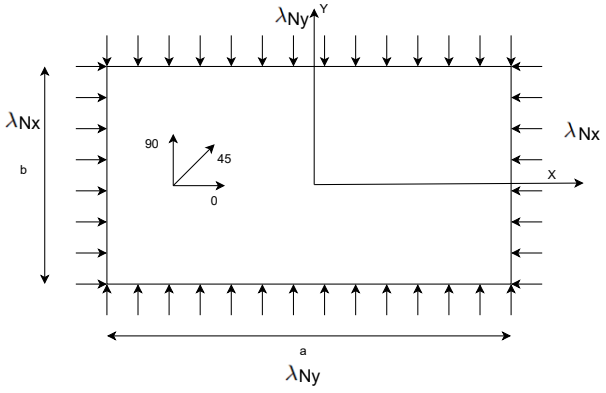


Figure 12: The geometry of laminated plate and applied loads (12).

Riche et al. (12) presents optimum benchmark cases for composite plates under bi-axial compression. As illustrated in Figure 12, N_x and N_y are the loads acting along the x and y axes, and λ represents a load amplitude parameter that affects all loads. The rectangular plates with length a and width b are simply supported. Discrete fiber angles of each layer are optimized, aiming minimum weight for a given design load, and the laminate is considered to have variable number of plies while remaining symmetric and balanced. It is assumed that only angles 0°, 90°, and ±45° can be used, and that all plies have the same thickness h_{ply}.

To account for a variable number of plies in this discrete optimization, the thickness value associated with each ply h_{ply}, becomes an additional design variable. Schläpfer (36) introduced the concept of ghost layers, which essentially is a layer that carries information about material properties and also the ply fiber orientation but can be associated with a zero thickness value. Here, every layer constituting the laminates are allowed to become ghost layers.

Two constraints are used in this optimization problem, represented by their respective load factors, being: critical buckling (λ_{cb}); and material failure (λ_{cs}). These are calculated as detailed in Appendix C. In the optimization, these constraints are normalized using the design load factor (λ_t), such that the optimization problem becomes:

$$\begin{aligned} \underset{\theta,t\in\mathcal{R}^n}{\text{minimize}} \quad & (\sum_{i=1}^{j} h_i)ab \\ \text{subject to} \quad & -MS_{cb} \leq 0 \\ & -MS_{cs} \leq 0 \end{aligned} \tag{31}$$

where $(\sum_{i=1}^{j} h_i)ab$ is the total volume of the laminate; defining the margins of safety for buckling MS_{cb} and failure MS_{cs} as:

$$\begin{aligned} MS_{cb} &= (\lambda_{cb}/\lambda_t) - 1 \\ MS_{cs} &= \lambda N_y = (\lambda_{cs}/\lambda_t) - 1 \end{aligned} \tag{32}$$

A genetic algorithm (GA) optimizer is selected given the discrete nature of the optimization of Equation 31. Adopting the ghost layer approach, the thickness of each ply can either be zero or h_{ply}, and each ply angle can assume one of the three values; 0°, 90° or ±45°. For implementation, the GA present within the Python module pymoo (37) is utilized. The deflation constraint herein proposed is applied to the optimizer to support the claim that the developed methodology is applicable in any optimization scheme that supports inequality constraints, even heuristic optimization schemes based on discrete variables. For this optimizer, the conversion from continuous to discrete variables can be done within the optimization, directly affecting the value of the objective function. In gradient-based methods this conversion is done after the optimization process is complete, in a post-processing step, usually affecting the objective and constraints. Regarding the main GA parameters, the population size herein utilized is 20, and the number of generations is 30, being the termination criterion.

Table 1 shows 4 stacking sequences with 48 plies with their corresponding load factors, extracted from Riche et al. (12). Their optimization focused on maximizing the buckling and failure load factors for a fixed amount of plies. Note the lowest load factor of 13,518.66 corresponding to the failure constraint. Here, with the intent to allow the optimizer to remove some layers, the design load factor is set to $\lambda_t = 10,000$. A total of 24 independent and discrete design variables are utilized, being 12 for the ply angles and 12 for the thickness of each ply. Note that due to the assumption of symmetric and balanced laminated, this setup will allow a maximum of 48 plies in the laminate.

Stacking sequence	Load factor	
	Buckling	Failure
$(90_2, \pm45_4, 0_4, \pm45, 0_4, \pm45, 0_2)_s$	14,168.12	13,518.66
$(\pm45_3, 0_2, \pm45_2, 0_2, 90_2, 0_4 \pm 45, 0_2)_s$	14,134.76	13,518.66
$(90_2, \pm45_3, 0_2, \pm45, 0_2, \pm45, 0_4, \pm45, 0_2)_s$	14,013.71	13,518.66
$(\pm45_2, 0_2, \pm45_2, 90_2, 0_4, \pm45, 0_2, \pm45, 0_2)_s$	13,662.61	13,518.66

Table 1
Results for the buckling load maximization problem obtained by Riche et al. (12). (48 plies, $a = 20$ in • $b = 5$ in, $N_x = 1$ lb, $N_y = 0.125$ lb).

Stacking sequence	Plies	Margin of Safety	
		Buckling	Failure
$(90_4, 0_6, 90_2, \pm45_6)_s$	48	7.44%	15.85%
$(\pm45_3, 0_4, \pm45_2, 0_2, \pm45_2, 0_2)_s$	44	6.49%	12.94%
$(\pm45_4, 0_4, \pm45, 0_2, \pm45, 0_2, \pm45)_s$	44	10.3%	12.94%
$(90_2, 0_6, 90_4, 0_2, 90_4, \pm45_3)_s$	48	3.84%	9.21%
$(90_4, 0_4 \pm 45_2, 90_2, 0_2, 90_4, 0_2, 90_2)_s$	48	13.73 %	5.155 %
$(90_2, \pm45, 0_2, \pm45, 0_2, \pm45_4, 0_4)_s$	44	2.55%	14.2%

Table 2
Results for the buckling load maximization problem obtained using the GA optimizer with deflation. (6 solutions generated using hypersphere approach with $p = 3$, $\sigma = 1$ and $r = 1$)

The solutions obtained are displayed in Table 2. The first solution is without deflation, resulting in a laminate with 48 plies, not changing from the baseline value. After deflation is applied, new constraints are added to the GA to enable finding novel solutions, where five new solutions are herein reported. The third solution is the most promising, combining the smallest amount of plies herein encountered with relatively high margins of safety.

Hence, applying deflation to this GA optimization proves that the deflation constraint works even in heuristic algorithms for optimization and it creates means to force further exploration of the design space.

3.2.2. Case study 2: deflation enabling gradient-based retrieval of fiber angles

Setoodeh et al. (39) discuss a method for generating curvilinear fiber paths for the manufacturing of VS laminates using AFP, where part of the work consists of finding the fiber angles when the lamination parameters (LP) are already known. Here, the deflation constraints are applied to enable the use of gradient-based methods to solve the multimodal optimization problem of retrieving the fiber angles.

A balanced and symmetric layup is assumed, and the objective function is given by:

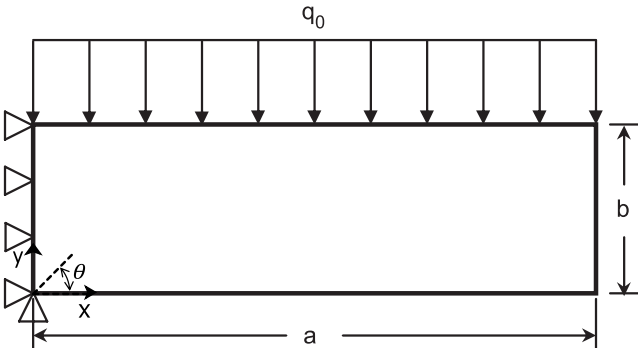


Figure 13: Cantilever plate with uniform load (38).

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & |V - V^*| \\ \text{subject to} \quad & 0° \leqslant \theta \leqslant 90° \end{aligned} \tag{33}$$

which minimizes the least square distance between the known LPs V^* and those calculated V based on the fiber angles being optimized. The calculated LPs use the relations in Equation D.7. The optimization of Equation 33 is already non-convex even for a single design variable, as shown in Figure 14.

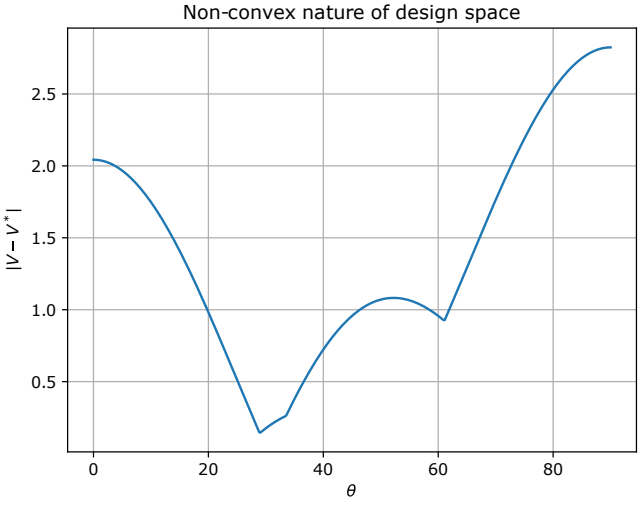


Figure 14: Variation of the objective function plotted against θ for a node along the laminate

The optimal set of lamination parameters V^* is found through a compliance minimization problem, where the feasibility constraints of LPs allow for a convex design space according to Fukunaga and Sekune (25; 26). The minimization problem is formulated as:

Method used	Normalized compliance $\bar{C}^*$	Normalized compliance $\bar{C}$	%Difference
Baseline from reference (39)	0.0374	0.0389	4.00
Gradient-based optimizer	0.0389	0.0414	6.42
Genetic Algorithm	0.00392	0.0419	6.88

Table 3
Normalized compliance obtained with optimum LPs and retrieved stacking sequence.

$$\begin{aligned} \underset{V_i}{\text{minimize}} \quad & \tfrac{1}{2} N_i^T \cdot A^{-1}(V_i) \cdot N_i \\ \text{subject to} \quad & 2V_1^2\left(1 - V_3\right) + 2V_2^2\left(1 + V_3\right) + \\ & V_3^2 + V_4^2 - 4V_1V_2V_4 \leqslant 1 \\ & V_1^2 + V_2^2 \leqslant 1 \\ & -1 \leqslant V_3 \leqslant 1 \end{aligned} \tag{34}$$

where N_i is the vector of the resultant forces for the i^{th} node; A is the in-plane stiffness matrix which is a function of V_i calculated with Equation D.8; and $V_i = \{V_{i_1}, V_{i_2}, V_{i_3}, V_{i_4}\}$, the vector of in-plane LPs of the i^{th} node. For balanced VS laminates, LPs $V_2 = V_4 = 0$. The material properties used for the present numerical evaluations are: $E_{11} = 181.0GPa$, $E_{22} = 10.3\,GPa$, $G_{12} = 7.17\,GPa$, and $\nu_{12} = 0.28$.

The results for the retrieved fiber angles can be verified by plotting the distribution of LPs once again and comparing it with the optimal LPs. In this case a non-dimensional compliance term is also utilised to numerically compare the results of the retrieved angles and optimal LPs. The non-dimensional compliance term is given as (38):

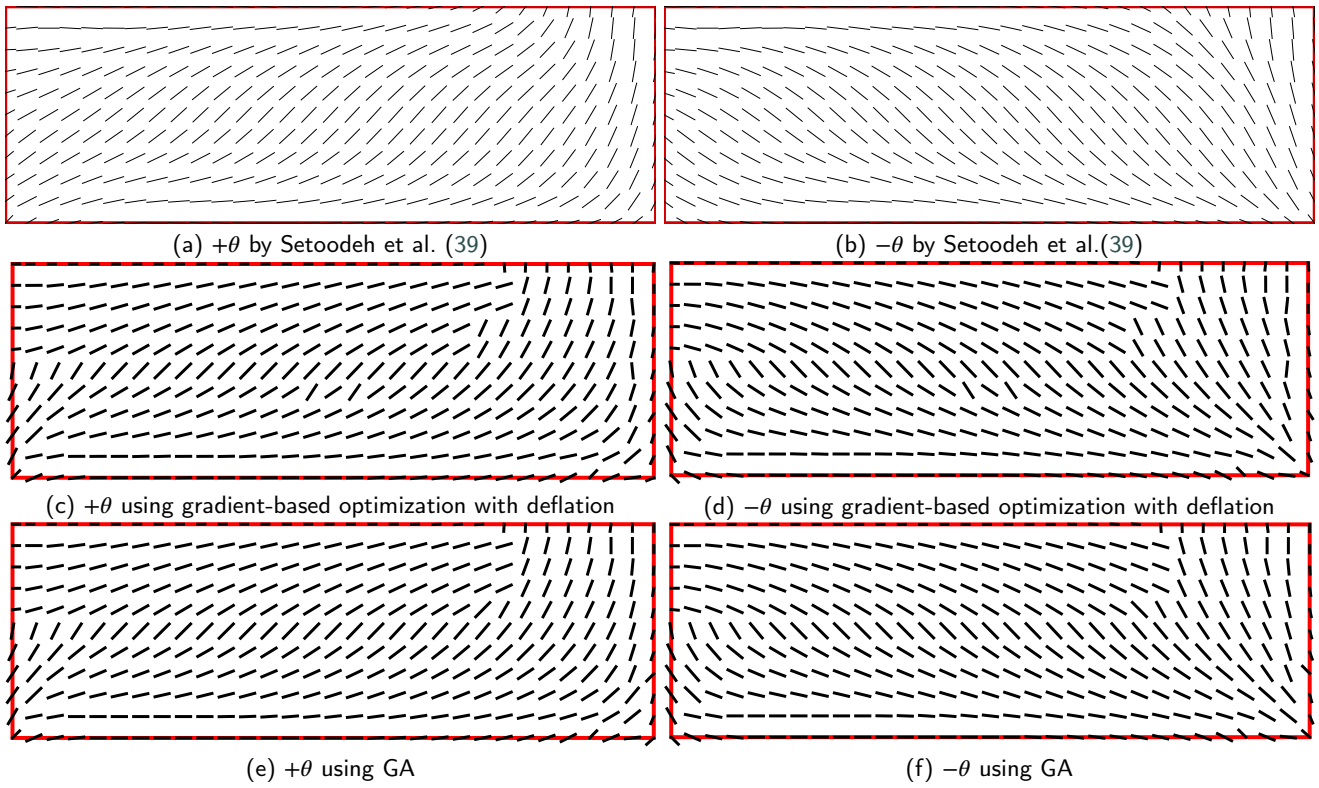
$$\bar{C} = \frac{E_{22} h b^3 C}{q_0^2 a^5} \tag{35}$$

where E_2 is the transverse modulus of elasticity; h, a and b are respectively the thickness, length and width of the laminate; q_0 is the uniformly distributed load; and C is the compliance obtained after optimization, given as:

$$C = \frac{1}{2} F^T \cdot U \tag{36}$$

where F and U are respectively the vectors of external forces and displacements.

For the sake of benchmarking, a standard GA algorithm (37) is also used to retrieve the fiber angles. Table 3 shows normalized compliance values obtained with the optimum LPs ($\bar{C}^*$); and with the retrieved fiber angles ($\bar{C}$). For the baseline result, Setoodeh et al. (39) achieved fiber angles that led to just 4% difference in compliance, and this is because the authors utilized a maximum curvature constraint for the curved fibers of ($\kappa = 3.333m^{-1}$) that further constrained the minimum compliance optimization of Equation 34. Figure 15 illustrates the fiber angle distributions corresponding to Table 3.



(a) $+\theta$ by Setoodeh et al. (39) (b) $-\theta$ by Setoodeh et al.(39)

(c) $+\theta$ using gradient-based optimization with deflation (d) $-\theta$ using gradient-based optimization with deflation

(e) $+\theta$ using GA (f) $-\theta$ using GA

Figure 15: Distribution of fiber angles of balanced laminate. ($a/b = 3$, and 31×11 nodes)

The results for the gradient-based method using the deflation constraint are obtained after deflating the design space 7 times, such that a total of 8 solutions are obtained after finding the retrieved angles that best minimizes the objective function of Equation 33. In Table 3, it can be seen that the percentage difference between the non-dimensional compliances obtained with V^* and V is approximately 6.68%, achieving a slightly better result than the benchmark GA that has a difference of 6.88%.

Figure 16 shows the distributions of laminations parameters obtained with the gradient-based method constrained by deflation, whereas Figure 17 shows these distributions obtained with the GA. Again, the good agreement shows that the deflation constraints enabled global search by the gradient-based optimizer. Figure 18 shows the convergence of the normalized compliance obtained obtained with each optimization, after progressive deflation steps, indicating convergence towards a global optimum.

4. Conclusion

The present study proposed a novel set of deflation constraints applicable to gradient-based and heuristics-based optimization algorithms. Virtually, the proposed constraints can be used in any optimization algorithm that is compatible with constraints. Deflation constraints force the optimizer to look for other optimal points, even if the optimization is restarted from the same initial guess. Such behavior makes it even possible for gradient-based optimizers to explore multimodal design spaces and ultimately find a global minima, after an enough number of deflation constraints are progressively added.

The three deflation constraint schemes herein proposed: hypersphere, hypercube, hypercuboid; were demonstrated in detail for a double-cosine function. Thereafter, two case studies related to composite design and optimization were investigated, where the first case study demonstrated the minimization of the weight of a plate for a given target design load, in which deflation is used to create new constraints for a GA algorithm. The second case study covered the optimization of LPs followed by the retrival of fiber angles, with deflation applied only to the gradient based optimizer. The obtained results were promising and showed the versatility of the developed methodology, uncovering new paths for its application composite design.

Future studies could focus in design cases involving multi-objective optimizations, investigating how the use of deflation constraints could improve the Pareto fronts obtained with gradient-based optimizers.

A. Deflation constraint - KKT proof

This appendix provides a proof on how the modified problem, after the introduction of the deflation constraint shown in Equation 10, provides a solution (x^*, y^*) that also satisfies the stationary conditions of the original problem. The modified problem is given as (33):

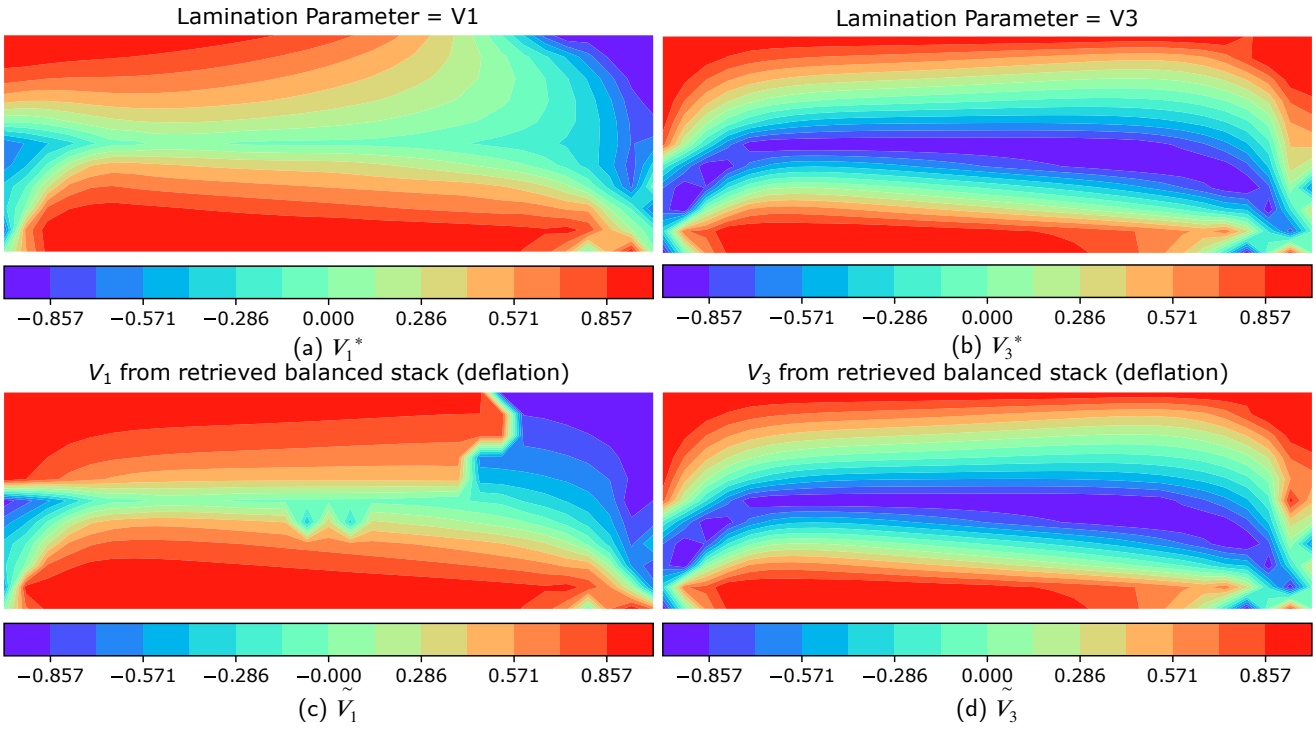


Figure 16: Distribution of Lamination parameters obtained by applying gradient-based optimization method with upper images representing optimum LP values and lower distributions obtained with retrieved angles shown in Figure 15 (c) and (d). ($a/b = 3$, and 31×11 nodes)

$$\begin{aligned} \underset{x\in\mathcal{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) = 0 \\ & m(x; x_1) \leq y \\ & l \leq x \leq u \end{aligned} \tag{A.1}$$

It is to be proved that, x^* is a regular KKT point to the problem given in Equation 7 and that $x^* \neq x_1$. Let a new Lagrange multiplier η be introduced in addition to the ones in Equation 8 associated with the deflation constraint. Therefore, the stationary conditions now will be:

$$\begin{gathered} \nabla_x f(x) + \nabla_x c(x)^T \lambda + z_+ + z_- + \eta \nabla_x m(x; x_1) = 0 \\ \eta = 0 \end{gathered} \tag{A.2}$$

It can be said that η would be 0 at any KKT point, hence the stationary conditions for Equation 7 would be satisfied. Since the constraints given in Equation 7 are a subset to those in Equation A.1, x^* must be a feasible point in the original problem (Equation 7) as well, and the complementary conditions of those constraints must be satisfied.

Since the obtained y^* is finite, (x^*, y^*) is feasible to the deflation constraint problem in Equation A.1 and m is bounded from below, then $m(x^*; x_1)$ must be finite, which implies that $(x^* \neq x_1$. This proof can also be generalized to inequality-constrained NLPs, hence the proposed deflation constraint approach is a generic and non-invasive way of utilizing the deflation method for an optimization problem (33).

B. Using last optimum as starting point

After a new deflation constraint is added, the optimization is restarted using as start point the last found minimum, which resulted in a decrease of the number of deflations required to locate all minima points of the double-cosine function. Figures 19, 20 and 21 show the evaluated points.

C. Buckling and failure constraints

The critical buckling load factor can be calculated with the following analytical expression (12):

$$\frac{\lambda_b(m,n)}{\pi^2} = \frac{D_{11}(\frac{m}{a})^4 + 2(D_{12} + 2D_{66})(\frac{mn}{ab})^2 + D_{22}(\frac{n}{b})^4}{(\frac{m}{a})^2 N_x + (\frac{n}{b})^2 N_y} \tag{C.3}$$

where m,n are respectively the half waves along the length and width of the plate; and D_{ij} are the components of the bending stiffness matrix $\boldsymbol{D}$ of the laminate, from the classical laminated plate theory (1). The values of m and n need to be found such that it minimizes λ_b to yield the critical buckling load λ_{cb}; and they vary with the number of plies, constitutive properties, plate dimensions and the load case.

The material failure constraints are given by the principal allowable strains for each ply. The calculation of the

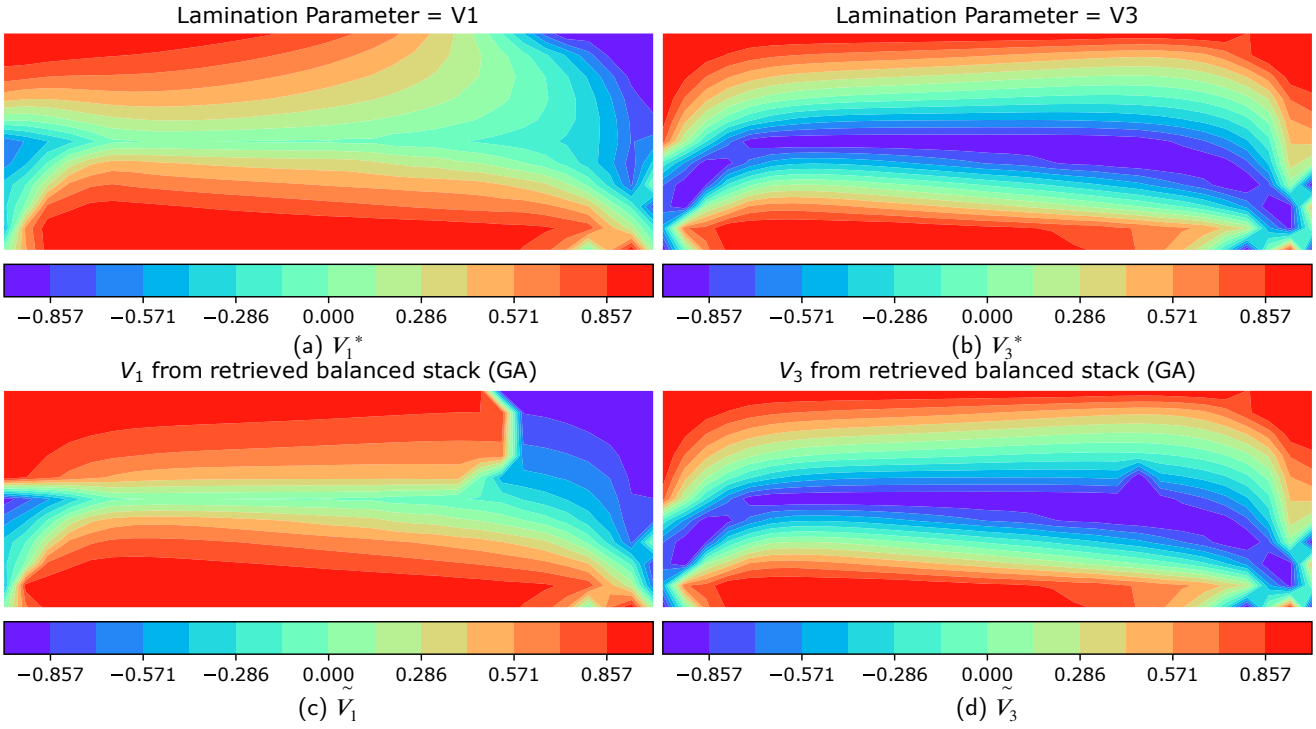


Figure 17: Distribution of Lamination parameters obtained by applying GA optimization method with upper images representing optimum LP values and lower distributions obtained with retrieved angles shown in Figure 15 (e) and (f). ($a/b = 3$, and 31×11 nodes)

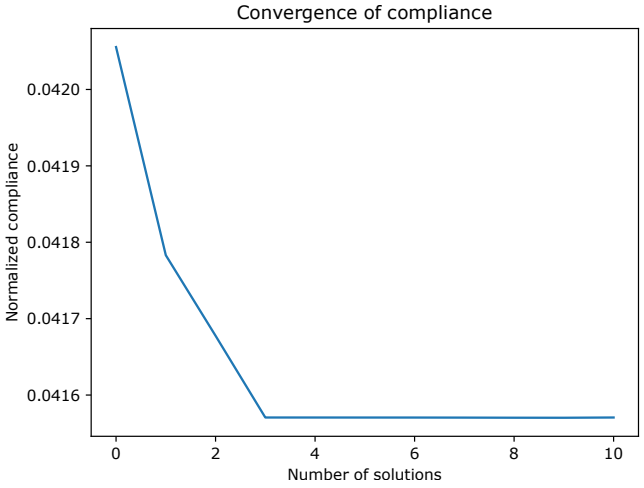


Figure 18: Convergence of Normalized Compliance with increase in number of solutions for deflation

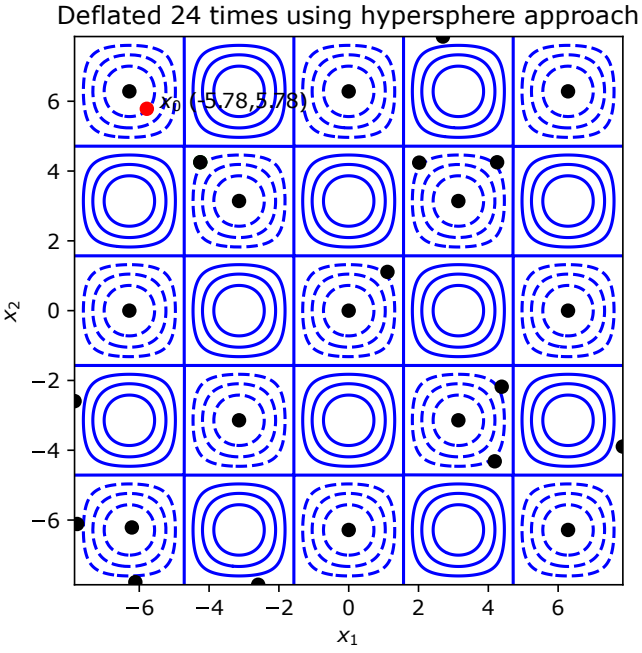


Figure 19: Hypersphere approach applied to locate minima of the double-cosine function (Equation 29) with updated start points

principal strains are done using the following relations:

$$\begin{aligned} \lambda N_x &= A_{11}\epsilon_x + A_{12}\epsilon_y \\ \lambda N_y &= A_{12}\epsilon_x + A_{22}\epsilon_y \end{aligned} \tag{C.4}$$

$$\begin{aligned} \epsilon_1^i &= \cos^2\theta_i\epsilon_x + \sin^2\theta_i\epsilon_y \\ \epsilon_2^i &= \sin^2\theta_i\epsilon_x + \cos^2\theta_i\epsilon_y \\ \gamma_{12}^i &= \sin^2\theta_i(\epsilon_y - \epsilon_x) \end{aligned} \tag{C.5}$$

where θ_i refers to the fiber angle associated with the i^th layer, with ϵ_x and ϵ_y being the global strains of the plate. The term λ in Equation C.4 is the load amplitude and it becomes λ_{cs} for the lowest value of λ such that only one of the principal strains in one of the layers exceeds the allowable strain. In the case herein considered, γ_{xy} is not present because the layers

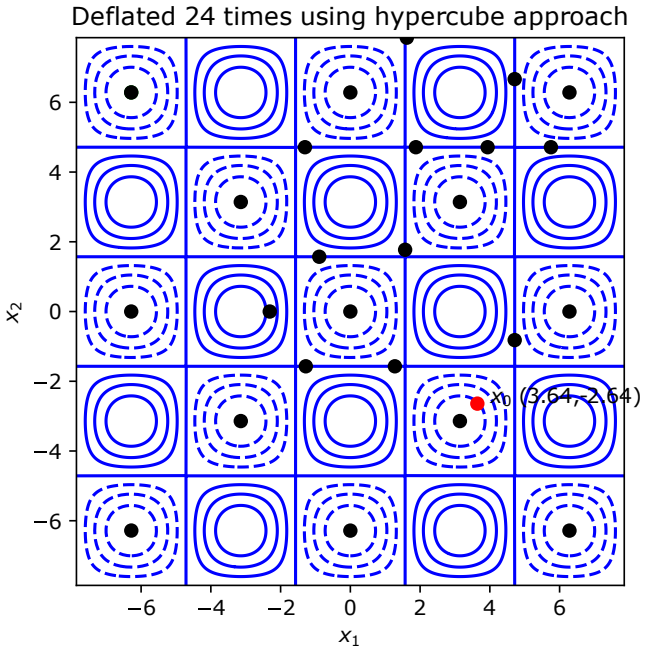


Figure 20: Hypercube approach applied to locate minima of the double-cosine function (Equation 29) with updated start points

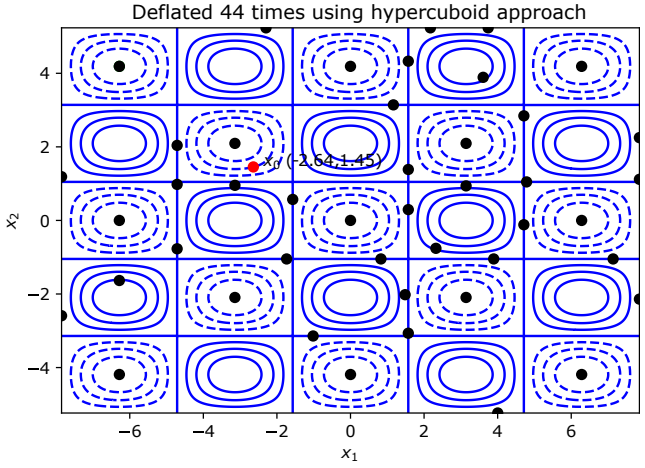


Figure 21: Hypercuboid approach applied to locate minima of the double-cosine function (Equation 29) with updated start points

of the laminate are always balanced and symmetric. The allowable strains for the adopted material, graphite epoxy composite, are given as:

$$\begin{aligned} \epsilon_1^i &\leq 0.008 \\ \epsilon_2^i &\leq 0.029 \\ \gamma_{12}^i &\leq 0.015 \end{aligned} \tag{C.6}$$

D. Lamination parameters

In this case study a few additional aspects need to be covered for enabling the reader's understanding. It is important to know the relation for lamination parameters to understand how it relates to the design variables and also the objective function. Lamination parameters were first introduced by Tsai et al. (23; 40) to represent the laminate layup configuration in a compact form. The transformation properties are derived using trigonometric relations (sines and cosines) in terms of multiple angles. The Lamination Parameters are non-dimensional and can in turn be used to obtain the in-plane and bending properties (ABD matrices) as they are related. A great advantage of using lamination parameters is that the number of design variables is reduced and is independent of the number of layers (20; 39). The 12 variables are given as:

$$\begin{aligned} \left(V_{1A}, V_{2A}, V_{3A}, V_{4A}\right) &= \int_{\frac{1}{2}}^{\frac{1}{2}} (\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta) dz, \\ \left(V_{1B}, V_{2B}, V_{3B}, V_{4B}\right) &= 4\int_{\frac{1}{2}}^{\frac{1}{2}} \bar{z}(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta) d\bar{z}, \\ \left(V_{1D}, V_{2D}, V_{3D}, V_{4D}\right) &= 12\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{z}^2(\cos 2\theta, \sin 2\theta, \cos 4\theta, \sin 4\theta) dz \end{aligned} \tag{D.7}$$

where V_{iA}, V_{iB} and V_{iD} are in-plane, coupling and flexural lamination parameters. $\bar{z}$ is the normalized z coordinate ($\bar{z} = z/h$ through the thickness and θ is the fiber orientation angle of that layer. The [A], [B], and [D] Matrices can be written as a linear function of the lamination parameters and the parameters of Tsai and Pagano as (23),

$$\begin{aligned} A &= h\left(\Gamma_0 + \Gamma_1 V_{1A} + \Gamma_2 V_{2A} + \Gamma_3 V_{3A} + \Gamma_4 V_{4A}\right) \\ B &= h^2/4\left(\Gamma_1 V_{1B} + \Gamma_2 V_{2B} + \Gamma_3 V_{3B} + \Gamma_4 V_{4B}\right) \\ D &= h^3/12\left(\Gamma_0 + \Gamma_1 V_{1D} + \Gamma_2 V_{2D} + \Gamma_3 V_{3D} + \Gamma_4 V_{4D}\right) \end{aligned} \tag{D.8}$$

where the material invariant matrices Γ_i's are given by the parameters of Tsai and Pagano which are material invariants.

CRediT authorship contribution statement

Sankalp S. Bangera: Methodology, investigation, validation, formal analysis, software, writing - original draft. **Saullo G. P. Castro:** Conceptualization, supervision, methodology, software, writing - original draft, data curation.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author, S.G.P. Castro, upon reasonable request.

References

[1] C. Kassapoglou, *Design and Analysis of Composite Structures*. Oxford, UK: John Wiley & Sons Ltd, 5 2013.

[2] Z. Wang, J. H. S. Almeida, A. Ashok, Z. Wang, and S. G. P. Castro, "Lightweight design of variable-angle filament-wound cylinders combining kriging-based metamodels with particle swarm optimization," *Structural and Multidisciplinary Optimization*, vol. 65, Apr. 2022.

[3] H. Ghiasi, D. Pasini, and L. Lessard, "Optimum stacking sequence design of composite materials part i: Constant stiffness design," *Composite Structures*, vol. 90, p. 1–11, Sept. 2009.

[4] H. Ghiasi, K. Fayazbakhsh, D. Pasini, and L. Lessard, "Optimum stacking sequence design of composite materials part ii: Variable stiffness design," *Composite Structures*, vol. 93, p. 1–13, Dec. 2010.

[5] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *J. Glob. Optim.*, vol. 56, pp. 1247–1293, July 2013.

[6] J. R. R. A. Martins and S. A. Ning, *Engineering design optimization*. Cambridge University Press, 2021.

[7] A. Rakhlin, O. Shamir, and K. Sridharan, "Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization," *arXiv preprint*, 9 2011.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint*, 9 2016.

[9] S. Kucherenko and Y. Sytsko, "Application of Deterministic Low-Discrepancy Sequences in Global Optimization," *Computational Optimization and Applications*, vol. 30, pp. 297–318, 3 2005.

[10] P. Kaelo and M. M. Ali, "Some Variants of the Controlled Random Search Algorithm for Global Optimization," *Journal of Optimization Theory and Applications*, vol. 130, pp. 253–264, 12 2006.

[11] R. M. Taylor, B. Niakin, S. Berdote, J. Deslich, Z. McFarlane, and A. M. Pankonien, "Optimization driven stiffness scaling and experimental validation of printed polymeric aeroelastic models," in *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2021.

[12] R. Le Riche and R. T. Haftka, "Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm," *AIAA Journal*, vol. 31, no. 5, pp. 951–956, 1993.

[13] S. Nagendra, D. Jestin, Z. Gürdal, R. Haftka, and L. Watson, "Improved genetic algorithm for the design of stiffened composite panels," *Computers amp; Structures*, vol. 58, p. 543–555, Feb. 1996.

[14] G. Soremekun, Z. Gürdal, R. Haftka, and L. Watson, "Composite laminate design optimization by genetic algorithm with generalized elitist selection," *Computers amp; Structures*, vol. 79, p. 131–143, Jan. 2001.

[15] C. A. C. António, "A hierarchical genetic algorithm with age structure for multimodal optimal design of hybrid composites," *Structural and Multidisciplinary Optimization*, vol. 31, p. 280–294, Jan. 2006.

[16] G. F. Gomes, S. S. da Cunha, and A. C. Ancelotti, "A sunflower optimization (sfo) algorithm applied to damage identification on laminated composite plates," *Engineering with Computers*, vol. 35, p. 619–626, May 2018.

[17] J. M. Van Campen, C. Kassapoglou, and Z. Gürdal, "Generating realistic laminate fiber angle distributions for optimal variable stiffness laminates," *Composites Part B: Engineering*, vol. 43, pp. 354–360, 3 2012.

[18] G. Clancy, D. Peeters, V. Oliveri, D. Jones, R. M. O'Higgins, and P. M. Weaver, "A study of the influence of processing parameters on steering of carbon Fibre/PEEK tapes using laser-assisted tape placement," *Compos. B Eng.*, vol. 163, pp. 243–251, Apr. 2019.

[19] B. C. Kim, P. M. Weaver, and K. Potter, "Manufacturing characteristics of the continuous tow shearing method for manufacturing of variable angle tow composites," *Compos. Part A Appl. Sci. Manuf.*, vol. 61, pp. 141–151, June 2014.

[20] M. A. Albazzan, R. Harik, B. F. Tatting, and Z. Gürdal, "Efficient design optimization of nonconventional laminated composites using lamination parameters: A state of the art," 2 2019.

[21] J. Foldager, J. S. Hansen, and N. Olhoff, "A general approach forcing convexity of ply angle optimization in composite laminates," *Structural Optimization*, vol. 16, p. 201–211, Oct. 1998.

[22] M. Miki, *Design of Laminated Fibrous Composite Plates with Required Flexural Stiffness*, p. 387–400. ASTM International100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959, Jan. 1985.

[23] S. W. Tsai and N. J. Pagano, "Invariant properties of composite materials," tech. rep., Air Force Materials Lab Wright-Patterson AFB OH, 1968.

[24] M. Picchi Scardaoni and M. Montemurro, "Convex or non-convex? on the nature of the feasible domain of laminates," *European Journal of Mechanics - A/Solids*, vol. 85, p. 104112, Jan. 2021.

[25] H. Fukunaga and H. Sekine, "Stiffness design method of symmetric laminates using lamination parameters," *AIAA Journal*, vol. 30, p. 2791–2793, Nov. 1992.

[26] H. Fukunaga and H. Sekine, "A laminate design for elastic properties of symmetric laminates with extension-shear or bending-twisting coupling," *Journal of Composite Materials*, vol. 28, p. 708–731, May 1994.

[27] J. H. Wilkinson, *Rounding errors in algebraic processes*. Dover Publications, 1994.

[28] P. Deuflhard, *Newton Methods for Nonlinear Problems*, vol. 35. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[29] P. E. Farrell, A. Birkisson, and S. W. Funke, "Deflation Techniques for Finding Distinct Solutions of Nonlinear Partial Differential Equations," *SIAM Journal on Scientific Computing*, vol. 37, pp. A2026–A2045, 1 2015.

[30] K. M. Brown and W. B. Gearhart, "Deflation techniques for the calculation of further solutions of a nonlinear system," *Numerische Mathematik*, vol. 16, pp. 334–342, 1 1971.

[31] J. Xia, P. E. Farrell, and S. G. Castro, "Nonlinear bifurcation analysis of stiffener profiles via deflation techniques," *Thin-Walled Structures*, vol. 149, p. 106662, 4 2020.

[32] P. E. Farrell, M. Croci, and T. M. Surowiec, "Deflation for semismooth equations," *Optimization Methods and Software*, vol. 35, pp. 1248–1271, 11 2020.

[33] M. Tarek and Y. Huang, "Simplifying deflation for non-convex optimization with applications in Bayesian inference and topology optimization," 2022.

[34] S. Bangera, *Global optimization using a deflation-based method for the design of composite structures*. Delft University of Technology, Faulty of Aerospace Engineering. MSc thesis. Published date: 23rd of March, 2023.

[35] N. Kyurkchiev and S. Markov, *Sigmoid Functions Some Approximation and Modelling Aspects: Some Moduli in Programming Environment Mathematica*. LAP LAMBERT Academic Publishing, 08 2015.

[36] B. Schläpfer and G. Kress, "A sensitivity-based parameterization concept for the automated design and placement of reinforcement doublers," *Composite Structures*, vol. 94, pp. 896–903, 2 2012.

[37] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[38] S. Setoodeh, M. M. Abdalla, and Z. Gürdal, "Design of variable-stiffness laminates using lamination parameters," *Composites Part B: Engineering*, vol. 37, pp. 301–309, 6 2006.

[39] S. Setoodeh, A. W. Blom, M. M. Abdalla, and Z. Gürdal, "Generating curvilinear fiber paths from lamination parameters distribution," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 5, pp. 3440–3452, American Institute of Aeronautics and Astronautics Inc., 2006.

[40] S. W. Tsai and H. T. Hahn, *Introduction to Composite Materials*. Routledge, 5 2018.