

Rapid Transformation of Lamination Parameters into Stacking Sequences

Moritz Sprengholz, Hendrik Traub*, Michael Sinapius

*Technische Universität Braunschweig
Institute of Mechanics and Adaptionics
Langer Kamp 6, 38106 Braunschweig, Germany*

Sascha Dähne, Christian Hühne

*DLR German Aerospace Center
Institute of Composite Structures and Adaptive Systems
Lilienthalplatz 7, 38108 Braunschweig, Germany*

Abstract

Lamination parameter optimisation is a highly efficient type of composite optimisation. An equally efficient transformation of lamination parameters into stacking sequences is not yet available. This paper presents a general method for rapidly transforming lamination parameters into continuous stacking sequences. Systematic studies of the relationship between lamination parameters and stacking sequences provide a broad understanding of the transformation problem. An important finding is that multiple stacking sequences share the same lamination parameter set. The transformation is therefore not unique and has to account for multiple layup solutions. The layup retrieval algorithm uses primitive optimisation techniques to search for all optima in the layup space. Admissible ply angles and layer numbers are hereby not restricted. In two representative examples, the authors show the algorithm's capabilities of finding all stacking sequence solutions of a twelve layer laminate and of finding multiple stacking sequence solutions for arbitrary layer numbers. This makes the algorithm applicable for stacking sequence retrieval, the last step in lamination parameter optimisation.

Keywords: Lamination parameters, Composite materials, Ply angles, Stacking sequence, Optimisation, Layup retrieval

1. Introduction

Weight optimisation of composite structures is a major challenge in lightweight design. Xu et al.[1] and Nikbakt et al[2]. review different approaches to optimise

*Corresponding author: h.traub@tu-braunschweig.de

composite structures, such as gradient-based optimisation using lamination parameter formulation. Lamination parameters define the stiffness properties of a laminate, independent of the number of plies and the underlying material. Twelve lamination parameters are sufficient to fully describe any possible laminate stiffness design. Lamination parameters can be optimised using gradient-based algorithms because of their continuous definition and the convexity of their feasible domain [3, 4]. Scardaoni et al. [5] recently have shown that the design space is indeed not convex and therefore smaller than previous convex approximations. The unique possibility of using gradient-based algorithms and the limited number of optimisation parameters identifies lamination parameters as an ideal tool for the rapid optimisation of composite structures [6]. It is also the only way of finding the global optimum of the stiffness design.

The manufacture of composite structures depends on detailed information about the laminate’s layup, defined by layer angles and their sequence in the laminate. Retrieving this information from lamination parameters is an ongoing challenge. Early retrieval approaches used genetic algorithm and particle swarm optimisation [7]. Recently, Viquerat has shown that for a twelve-ply laminate with a defined set of lamination parameters, there are 112 exact layup solutions [8]. Liu et al. used a branch and bound algorithm to approximate a single layup solution for a given set of admissible ply angles [9]. Liu’s algorithm approximates layups for an arbitrary number of plies, but only allows a fairly restricted set of ply angles. Viquerat on the other hand allows fully continuous ply angles but is restricted to a twelve layer laminate. In contrast to both approaches, engineering applications demand a broad range of layer numbers and ply angles.

Appropriate solutions for layup retrieval must be able to cover both, a broad range of layer numbers and fine resolution of layer angles. Applicable layup retrieval solutions need to be able to cover both a broad range of layer numbers, and a fine layer angle resolution. Multiple solutions are beneficial because they allow the implementation of additional restrictions. For one set of lamination parameters, multiple solutions permit to consider manufacturing and continuity constraints without affecting the original optimisation results.

This paper proposes a rapid layup retrieval algorithm using basic optimisation techniques. A detailed study of the optimisation problem reveals the actual target of the layup retrieval algorithm, i.e. to find as many local optima as possible. Many of these local optima represent exact solutions, since many layups have identical stiffness properties. All local optima, representing exact solutions, are at the same time global optima in the sense that no lower values can be found in the optimisation problem. Therefore, unlike other approaches, the proposed algorithm tries to find as many local optima as possible. The algorithm retrieves multiple exact solutions per minute, using only primitive optimisation techniques and stopping immediately at each local optimum.

2. The relationship between layup and lamination parameters

Lamination parameters depend on the layer angles, the layer thicknesses and the stacking sequence. While the calculation of the lamination parameters is

trivial, retrieving the laminate’s layup from them is not. The retrieval problem can be studied by manipulating the underlying equations, or by working with simple examples. Examples of laminates with few layers ($N \leq 3$) allow to visualise the effect of layer angles on lamination parameters. The visualisation provides insights into feasible layup solutions for a given set of lamination parameters. This section studies the transformation problem using both example visualisation and equation analysis.

2.1. Characterising parameter spaces

Commonly, ply angles are directly modified in stiffness based composite optimisation (fig. 1). Geometrically, each angle θ_n represents a coordinate axis in an N -dimensional Cartesian space. Every stacking sequence with N plies occupies a unique point inside this space, which we will refer to as layup space from here on.

Each angle θ_n can take values on the interval of $(-90^\circ; 90^\circ]$. As both ends of the interval represent the same ply orientation, it is more appropriate to consider it as a circular interval. In other words, the layup space wraps around on itself at its ends. Any angle combination is a valid layup. Therefore the admissible angles in all layers are uniformly distributed. This builds the basis of the layup space.

The CLT’s¹ [10] ABD matrix represents the laminate’s stiffness and can be calculated from its layup. The optimisation of the ABD reduces the optimisation parameters to a constant number. However, the entries of the ABD matrix are not independent. The 12 independent parameters fully describing the stiffness properties of the ABD matrix are the lamination parameters. Thus, analogous to layup space, the LP space exists. There the twelve LPs span a twelve dimensional space as its coordinate axes. Knowing where a point inside LP space is located in layup space solves the stacking sequence retrieval problem.

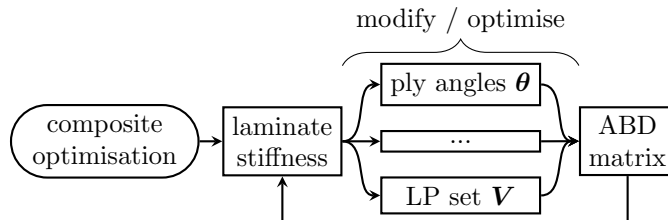


Figure 1: Different approaches to composite stiffness optimisation

The following equations transform the layup space into the LP space [11–13]. If all single plies are of the same material and thickness, the twelve lamination

¹Classical Lamination Theory

parameters V are defined by

$$\begin{aligned} V_x^A &= \frac{1}{N} \sum_{n=1}^N (z_{n+1} - z_n) W_x \\ V_x^B &= \frac{2}{N^2} \sum_{n=1}^N (z_{n+1}^2 - z_n^2) W_x \\ V_x^D &= \frac{4}{N^3} \sum_{n=1}^N (z_{n+1}^3 - z_n^3) W_x \quad . \end{aligned} \quad (1)$$

The trigonometric functions W_x are defined depending on x as

$$W_x = \begin{bmatrix} \cos 2\theta_n, & x = 1 \\ \sin 2\theta_n, & x = 2 \\ \cos 4\theta_n, & x = 3 \\ \sin 4\theta_n, & x = 4 \end{bmatrix} \quad . \quad (2)$$

The distance of each ply n to the laminate centre is defined by

$$z_n = -\frac{N}{2} + n \quad . \quad (3)$$

This leads to the function notation

$$f : (-90^\circ; 90^\circ]^N \rightarrow [-1; 1], \theta^N \rightarrow V^{12} \quad (4)$$

which describes the transition from layup into LP space. Each LP is constrained on the interval $[-1; 1]$, analog to the extreme values of the underlying trigonometric functions. Regardless of the number of plies N , only twelve scalar lamination parameters are returned. Therefore, transitioning into LP space causes information loss. A defined stacking sequence always calculates to the same set of LPs. In return, there could be multiple layups for one LP set, with or without different N . Therefore, no single function exists to describe the transition from LP space into stacking sequences.

The LPs are split into three groups analogous to the ABD matrix. V_x^A describe extensional stiffnesses, V_x^B bending-extension coupling stiffnesses and V_x^D bending stiffnesses [10]. Within the three LP groups, only the trigonometric functions W_x differ. The remaining terms scale the output of W_x and are therefore referred to as scaling factors s from here on. If everything is brought into the sums in eq. 1 the scaling factors read as

$$\begin{aligned} s_n^A(n, N) &= \frac{1}{N} (z_{n+1} - z_n) \\ s_n^B(n, N) &= \frac{2}{N^2} (z_{n+1}^2 - z_n^2) \\ s_n^D(n, N) &= \frac{4}{N^3} (z_{n+1}^3 - z_n^3) \quad . \end{aligned} \quad (5)$$

The scaling factors describe the influence of each ply position on the resulting LP value. This is shown for two different numbers of plies in fig. 2.

s^A is constant for each ply and only changes with N . Eq. 5 thus can be simplified to $s^A(N) = 1/N$. Each ply with angle θ_n has the same influence on the V_x^A regardless of its position in the stacking sequence. This is not the case for the other two groups. s^B changes linearly with the n -th ply. In the centre of the laminate, the influence is smallest. This is also where the sign of s^B changes. It is therefore the only scaling factor that can be negative. For s^D , there exists a quadratic relationship that provides a positive sign. The influence diminishes faster towards the centre of the laminate than in s^B .

For all scaling factors, the absolute value for each ply n becomes smaller as the number of plies N increases. Only in this way the interval bounds $[-1; 1]$ of the LP can be maintained. Therefore, the influence of each single ply on the resulting LP diminishes with growing N .

As the trigonometric functions in W_x are only scaled by the scaling factors, the extrema of W_x are unchanged in position but not in value. This relation is displayed for the first LP of each group over all possible angles in fig. 3. The trigonometric function W_1 is the same for all plots. As the plot is limited to $N = 2$, a few restrictions apply. The V_x^A are symmetric in both dimensions as $s_1^A = s_2^A$. This is also the case for higher dimensions. All s^B and s^D only have the same absolute values in the two-dimensional case. Negative s^B -factors result in the LP surface of V_1^B being shifted by 90° in θ_1 direction with respect to the V_1^A surface. As in this case all s^A and s^D are identical, the values of s^D have been changed to obtain s^{D^*} . s^{D^*} shows the effect of non-identical scaling factors found in higher dimensions ($N > 2$). The surface of $V_1^{D^*}$ is warped but not shifted.

In summary, changing the sign of a scaling factor s_n results in a shift of the surface along the corresponding θ_n axis, while a change in magnitude only distorts the surface. The shifting length matches half the period of the underlying trigonometric function W_x of each LP. This applies to all LPs.

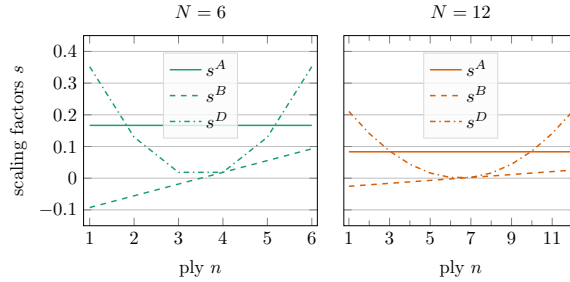


Figure 2: Scaling factors s for $N = 6$ and $N = 12$

2.2. Reversing the transformation

In the previous section, we partitioned the equations for the lamination parameters into scaling factors and trigonometric functions. This enables the

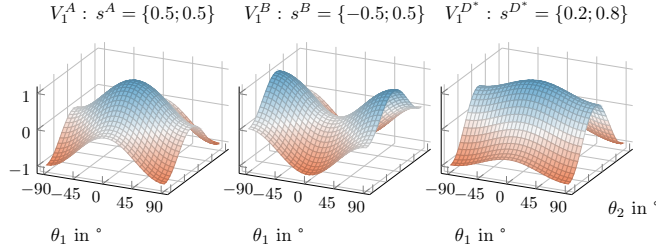


Figure 3: Influence on the LP solution surface of V_1^A , V_1^B and $V_1^{D^*}$ by the scaling factors s . Both axes are partitioned equally.

inversion of the function notation in eq. 4. The scaling factors are known, if the total number of plies of the stacking sequence is known. This is often the case as optimisation of composites using lamination parameters also returns an optimal total thickness. The individual ply thickness of the employed fabric determines the number of plies N . The scaling factors are independent of the ply angles we seek to retrieve. Inversion of the trigonometric functions is therefore the key step in reversing the transformation.

In the following, we consider solutions in low-dimensional space to predict positions of solutions for larger N . The simplest case is one-dimensional. The functions W_x are displayed in fig. 4 depending on a single angle θ . As the scaling factors are 1 if $N = 1$, all W_x directly describe the LPs. Furthermore, all LP groups (V_x^A , V_x^B , V_x^D) are identical.

To reverse the transformation, horizontal lines can be drawn in fig. 4 for the selected LP values. The intersection points between the line and the corresponding W_x mark the solution angles to that particular back transformation. There are two intersections with $\cos 2\theta$ and $\sin 2\theta$ and only one for extreme LP values. The number of intersections doubles for the trigonometric functions with halved periods.

Section 2.1 already shows that more than one solution is possible. Therefore, not one single inverse function exists over the entire domain θ . To obtain intervals in which all of W_x are always invertible the domain has to be split into eight parts of 22.5° . We further refer to this length as the partition length. This is necessary as all LPs and therefore all W_x depend on the same angles θ_n . They cannot be considered independently.

Considering two dimensions leads to many changes. For a more compact representation, we only present an exemplary LP set instead of the entire domain. The LP set of the $[45^\circ; 15^\circ]$ laminate is given in tab. 1. To obtain the individual LP solutions, the surfaces (as for example shown in fig. 3) can be intersected by a plane at height of the corresponding LP value. The intersection points can be displayed as a contour plot, see fig. 5. Since V_x^A and V_x^D are identical if $N = 2$, only 8 of the 12 LP solutions are plotted. In the left part of the figure, LPs with a period of 180° are plotted (W_1 , W_2). All LPs are drawn combined in the right part of the figure. The individual solutions, i.e.

the angles that produce one particular LP, are continuous one-dimensional rings instead of discrete points of the one dimensional case. Their radii depend on the LP value while their centres are determined by the trigonometric function. Depending on the periodicity of W_x , either one or four rings per LP.

The total solution is the intersection between all rings of each individual LP. It is marked by a black square and corresponds to the initial laminate. Noticeably, there are other points where multiple, but not all, rings intersect. The stacking sequence $[-45^\circ; 15^\circ]$ is the intersection of six out of eight rings (marked by a black triangle). Symmetry and periodicity to the total solution seem to determine the location of those apparent solutions. In contrast to the 1D case, generally no inverse functions exist in higher dimensions. In two dimensions, partitioning the domain cannot result in an inverse function. However, if we partition the domain along each axis, we find that the partition lengths seem to grow. While there are two discrete solution points per LP in one dimension for V_1^A ($\cos 2\theta$), only one ring in two dimensions exists. The two discrete solutions are merged into one ring in the higher dimension. Therefore, the partition lengths can be increased, but so does the dimension N , which ultimately leads to more partitioned fields.

Fig. 6 assists in understanding the influence of the LP value on the radius of the solution ring by example for V_1^A . The ring is largest if $V_1^A = 0$. It takes the shape of a rhombus. If V_1^A approaches the value 1 the ring shrinks and resembles more and more a circle until it vanishes in the centre point (marked by black cross). The same happens when approaching -1 but the centre point is shifted by half a period on both axes. Since both scaling factors s^A are identical, the rings are not distorted. Extreme LP shrink the shape on which solutions lie. Hence, such LP sets generally contain fewer total solutions.

The previous paragraphs have investigated inverse transformation in one and two dimensions. Three dimensions lead to generalizations that are made below. However, they are not presented further here. We introduce definitions below to build an abstraction for any number of dimensions.

The solution shape denotes the actual shape where a solution lies for the back transformation. This can be a point in 1D ($N = 1$), a line (or ring) in 2D ($N = 2$), and a surface in 3D for individual (per LP) solution shapes. In higher dimensions ($N > 3$), there are N -dimensional hyper surface solution shapes. The number of hyper surface solution shapes per LP is one for the first two LPs in each group (based on W_1 and W_2). Otherwise, there are 2^N of them per LP (fig. 2). The solutions shapes are distorted by their corresponding scaling factors.

It needs to be differentiated between solution shapes for one LP and a whole set. Therefore, there are individual solution shapes and total solution shapes. Total solution shapes have a dimension of $\max(0, N - 12)$ as they result by intersecting all individual solution shapes with each other. That means 12 ply laminates have total solutions of dimension zero, i.e. total solution points. If $N > 12$, the total solutions consist of infinitely many point solutions, i.e. a set of solutions, similar to an under-determined system of equations.

This raises the question of how best to apply stacking sequence retrieval in

V	1	2	3	4
V_x^A	0.433	0.75	-0.25	0.433
V_x^B	0.433	-0.25	0.75	0.433
V_x^D	0.433	0.75	-0.25	0.433

Table 1: LP set of $[45^\circ; 15^\circ]$ laminate in fig. 5

practical cases. The direct search for solutions for more than 12 layers cannot find all of them, since infinitely many exist. Selecting the optimal solution is therefore highly unlikely. In contrast, 12 plies allow to find all solutions², which then have to be scaled to the desired number of plies. This exchanges the accuracy of each ply angle for more plies. This also allows a reduction of admissible layer angles. We propose the term *layer-and-angle redundancy theorem* for this observation. However, an upscaling method is not part of this research.

A successful retrieval of laminate layups from LPs requires the equivalent of the one-dimensional inverse function in higher dimensions. If such an inverse function cannot be found, a numeric approach such as an optimisation is able to solve the transformation iteratively. Optimization methods converge reliably to one solution if at most one continuous solution shape exists within the set boundaries. Solution spaces therefore define subsets of the entire layup space where a solution is likely located. If infinite solutions exist inside a subspace, only a finite number of them may be found.

In the previous section, it has already been elaborated that the partition length increases in higher dimensions as solution shapes merge. As we cannot visualize the effect in higher dimensions, we propose to keep the 22.5° partition length as a conservative estimate.

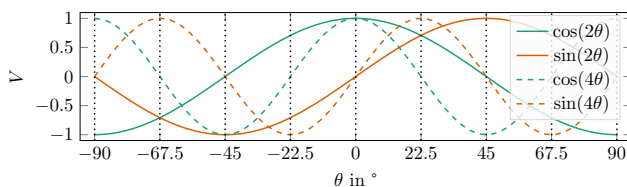


Figure 4: Plotting of all V is possible just via W_x if $N = 1$. To inverse all of the trigonometric functions at any given point the domain has to be partitioned into 22.5° steps (partition length).

Looking at solutions for stacking sequences of up to two plies, we have shown by example that higher N permit more stacking sequences for a given LP set.

²The stiffness distribution of each existing laminate of one material can be realised with 12 layers when arbitrary layer angles are allowed.

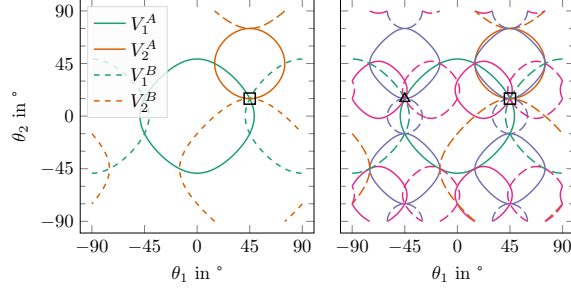


Figure 5: Individual solution shapes for all V (duplications omitted) of $[45^\circ; 15^\circ]$ laminate (LP set in tab. 1). The figure on the left displays only the V_1 and V_2 while all V are plotted on the right. The total solution (intersection of all V) is marked by a black square.

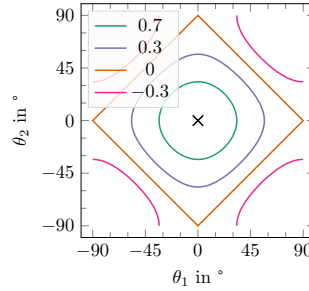


Figure 6: Shape of solution rings depending on the value of V_1^A . The centre point is marked by a black cross.



N	1	2	3	\dots	n
solution shapes	\longrightarrow			\dots	
#	2	4	8	\dots	2^n

Table 2: Identifying the number of individual solutions shapes per LP (V_3, V_4) relative to N . The solution shapes are portrayed as circles and spheres. See fig. 6 for an accurate representation of shape.

With rising N , the number of intersections between the solution shapes of each individual LP increase. Multiple stacking sequences sharing the same LP set are possible if $N \geq 12$. An infinite number of total solutions may be found if $N \geq 13$. As no analytical inverse function exists, solutions may be retrieved iteratively via optimisation.

3. The optimisation problem

In stacking sequence retrieval the optimization algorithm must be able to find multiple solutions. These are scattered in a large, high-dimensional search space which needs to be traversed efficiently. The optimisation domain is only constrained to the explicit boundaries $\boldsymbol{\theta} \in (-90^\circ; 90^\circ]^{N3}$.

The objective function compares the target LP set \mathbf{V}_t to the LP set \mathbf{V}_i of the current angles $\boldsymbol{\theta}_i$ via the *Root Mean Square Error* (RMSE):

$$\text{RMSE} = \sqrt{\frac{(\mathbf{V}_t - \mathbf{V}_i)^2}{12}} \quad . \quad (6)$$

It is easy to evaluate (see LP equations 1). Therefore, the algorithm itself also should be fast, even if it requires more function evaluations. The objective function is smooth due to its trigonometric origin (shown by example in fig. 7) and its gradients are available analytically.

Most often genetic algorithms, a type of global optimisation algorithms, are used for stacking sequence retrieval [7, 14]. Evolving a population of candidate solutions is not reasonable if multiple equally valid solutions exist. Applying genetic algorithms for stacking sequence retrieval is only viable when applied inside a small subspace. The same applies to surrogate modelling algorithms. Additionally they build a model which is expensive compared to evaluating the cheap objective function directly. They also suffer proportionally more in higher dimensions due to their complexity. Using multiple starts enables any algorithm to find multiple solutions. The application of a database consisting of precomputed pairs of LP sets and stacking sequences revealed extreme storage space requirements which prevents any practical use.

Due to the unique properties of the problem, none of the above mentioned algorithms seem to be a viable approach if used individually. We therefore propose to combine multiple of them into a multistart, multistep algorithm. This enables a fast and yet accurate back transformation.

4. A rapid, universal layup retrieval algorithm

The proposed rapid layup retrieval algorithm applies multistart optimisation algorithms especially suited for finding multiple minima. In stacking sequence

³If the boundaries are explicitly defined it must be ensured that they can wrap around on themselves during optimisation (see section 2.1). Otherwise optimisation can also be done without any restrictions. Then the solution $\boldsymbol{\theta}$ can be brought back into the valid domain via modulo operations.

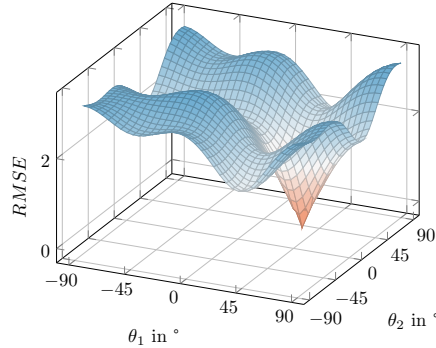


Figure 7: Plot of the RMSE optimisation surface for the $[45^\circ; 15^\circ]$ laminate (LP set in tab. 1)

retrieval, a smooth and high-dimensional problem, multiple local as well as global minima may exist. Finding the solution spaces surrounding these minima is paramount in building an efficient algorithm. Starting at multiple points, a multistep algorithm uncovers solution spaces and approaches global minima therein rapidly. This section presents the algorithm separated into three levels of fidelity in rising order.

Fig. 8 shows the first two steps of the algorithm by means of a graphical example in two dimensions. Coordinate descent begins at a generated start point as the first step (a). After coordinate descent has been repeated for all N coordinates, the gradient is estimated at the last and therefore best found angle combination (b). Line search is then performed once in negative gradient direction as the second step (c). The third step consisting of a local, highly accurate optimisation is not shown in fig. 8. The pseudocode of the algorithm (see Algorithm 1) builds the basis for the following paragraphs where the individual steps will be described and explained in detail.

The algorithm requires as main inputs a target LP set \mathbf{V}_t and a number of plies N to find solutions for. Additional parameters are a number of iterations i_{max} (respectively starting points), an optimisation threshold value ε and a step size $\Delta\theta$ for starting point generation. The layups array \mathbf{L} and optimisation residuals \mathbf{err} are initialized. Each iteration is independent and provides a unique starting point to the multistep algorithm. Starting points in layup space may be generated randomly or in a certain pattern, e.g. latin hypercube sampling in order to ensure space filling. The step size defines how far apart starting points are in each dimension. It should be upper-bounded by the partition length and thus never exceed 22.5° .

The first optimisation step⁴ performs coordinate descent with a constant step size $\Delta\theta_{cd}$. Only one coordinate changes at a time, usually starting from the first.

⁴This “step” describes a part of the multistep (i.e. multipart) algorithm. It should not be confused with the step size $\Delta\theta$, i.e. an increment.

The angle corresponding to the lowest objective function value (argument of the minimum) for each coordinate is stored. Only angles that lie on the grid defined by the step size are available for selection. This sequence is repeated for all N coordinates. The number of function evaluations is low, resulting in very fast execution times.

Coordinate descent evaluates the entire domain. When the coordinate reaches the end of the domain, it is wrapped around to the other end. Coordinate descent is not influenced by local or global minima as it does not evaluate the gradient of the objective function and the step size remains unchanged. It can therefore jump out of valleys or pass right through them. Yet, these design choices also deny approaching a minima that is nearby as this would require changing the step size and not limiting the search directions to the coordinate axes. Coordinate descent can also easily get stuck when the current angle already minimizes the objective function. In this case, no improvement or change is possible anymore. Coordinate descent clearly depends on the start value and is therefore well suited for a multistart approach.

To circumvent the movement restrictions of coordinate descent, a second step is introduced. In coordinate descent all search directions are perpendicular to each other. As it is unlikely that a solution sits directly on one of the coordinate grid points, more search directions should be permitted. Line search employs a rough and therefore smoothed gradient approximation (finite difference) to set a new search direction. Again, the entire space is evaluated in rough steps of $\Delta\theta_{l_s}$. The only thing differing from coordinate descent is how the search direction is selected.

Line search can be repeated multiple times. However, it is primarily used for restarting stuck coordinate descents, where one iteration is often sufficient. It has a slightly larger computational cost per search direction compared to coordinate descent, since the gradient has to be approximated. In practice, however, it is much lower as line search is often used only once (one search direction) compared to N search directions in coordinate descent.

Line search was initially developed to improve a solution locally where an accurate gradient is useful. In its current state it is used for its ability to quickly traverse the entire search space for even lower objective function values. It effectively restarts stuck coordinate descents, but due to its fixed step size, it is not necessarily suited for improving results locally.

Both coordinate descent and line search can be repeated individually and in combination. Quick trials for parameter selection have shown that performing coordinate descent three times along all N coordinates and then line search once yields good results. Repeating these steps in turn three times, reduces the objective function until no significant improvement can be achieved for more iterations⁵. The number of function evaluations thus remains low.

Finally, an accurate, local step remains to converge on a possible nearby

⁵Moving away from fixed numbers of iterations and instead utilising a stopping criterion based on no further improvement would be a useful addition to the algorithm.

solution. The RMSE being under some selected threshold value ε serves as an indicator for being inside a solutions space close to a total solution (global minima). As these solution spaces appear to be convex and smooth (fig. 7), a local search algorithm should find the total solution, if one exists. *Derivative-Free Optimizer for Least-Squares Minimization* (DFO-LS) [15] has been chosen as a gradient free algorithm, although *Matrix Adaptation Evolution Strategy* (MA-ES) [16] (a derivation of *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES)) has also been tried successfully. Gradient based optimisation algorithms have not been able to reach the same level of accuracy. If the residual of the optimisation is below a set value (usually 10^{-6}) it is safe to assume that a stacking sequence for the target LP set has been found.

Deriving the angle differences corresponding to the RMSE is not possible as the target stacking sequence is unknown. If the optimisation residual is small enough or manufacturing constraints limit the ply angle resolution, this problem can be neglected. Otherwise an upper and lower bound for the angle differences may be obtained by stochastic methods. Being able to control the error is one of the greatest strengths of this algorithm. It allows exchanging computation time against accuracy and therefore adapting to the requirements at hand.

The proposed algorithm is capable of returning several solutions to the stacking sequence retrieval problem per minute. It is independent from N and *embarrassingly parallel* in its design. The reference implementation utilises only a single thread for its computations and can thus be sped up significantly. The amount of found solutions depends on the LP set and N . Extreme LP sets are more difficult to solve because fewer solutions exist in general.

Algorithm 1 Layup retrieval algorithm

```

1: given  $\mathbf{V}_t \in \mathbb{R}_+^{12}$ ,  $N \in \mathbb{N}_+$ ,  $i_{max} \in \mathbb{N}_+$ ,  $\varepsilon \in \mathbb{R}_+$ ,  $\Delta\theta \in \mathbb{R}_+$ ,  $a = \lfloor \frac{180^\circ}{\Delta\theta} \rfloor$ 
2: initialize  $L \in \mathbb{R}^{i_{max} \times N}$ ,  $err \in \mathbb{R}^{i_{max}}$   $\triangleright$  Define layup and residual array
3: for  $i \leftarrow 0, \dots, i_{max} - 1$  do
4:    $\mathbf{z} \in \mathbb{R}^N \leftarrow \mathbf{U}(0, 1)$   $\triangleright$  Generate random start layup
5:    $L[i] \leftarrow \text{deg2rad}(\mathbf{z} \cdot (a - 1) \cdot \Delta\theta - 90^\circ + \Delta\theta)$ 
6:   for  $j \leftarrow 0, \dots, N - 1$  do  $\triangleright$  Coordinate descent
7:      $L[i][j] \leftarrow \text{getArgMinAlongCoordinate}(j)$ 
8:   end for
9:    $\mathbf{g} \leftarrow \text{estimateGradient}(L[i])$   $\triangleright$  Line search
10:   $L[i] \leftarrow \text{getArgMinAlongDirection}(-\mathbf{g})$ 
11:   $err[i] \leftarrow \text{getRMSE}(L[i], \mathbf{V}_t)$ 
12:  if  $err[i] \leq \varepsilon$  then  $\triangleright$  Optimisation
13:     $L[i], err[i] \leftarrow \text{DFO-LS}(L[i])$ 
14:  end if
15: end for
16: return  $L, err$   $\triangleright$  Return layups and residuals
```

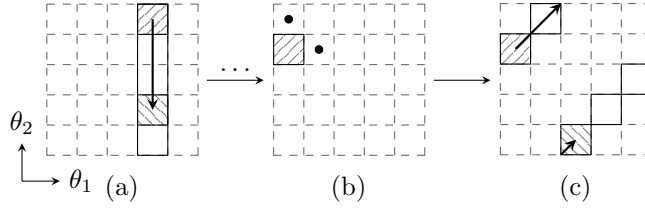


Figure 8: Graphical example of the layup retrieval algorithm in two dimensions. Only the first two steps coordinate descent (a) and line search (b and c) are shown.

5. Discussion and comparison of the layup retrieval algorithm

In engineering applications, the retrieval of laminate layups is subsequent to the LP optimisation. The optimisation of LPs often includes constraints such as a feasible domain in LP space [3]. A feasible domain offers the possibility to incorporate manufacturing constraints [17], e.g. the admissible ply angles are restricted to $0^\circ, \pm 45^\circ, 90^\circ$. LP sets resulting from an optimisation tend to be on the boundaries of their feasible domain. As described in section 2.2, such extreme laminates have far less layup solutions compared with an evenly distributed LP set, due to the significant restriction of the solution space. Therefore an applicable layup retrieval algorithm needs to be verified with sets of optimised lamination parameters.

The test of layup retrieval algorithms requires a set of optimised LPs. The optimisation of a simply supported CFRP panel results in such a set of LP used as the starting point in stacking sequence recovery. Figure 9 shows a simply supported CFRP panel including the outer dimensions and the applied loads. The uni-directional carbon fibre tape T-300 15k/976 [18] is selected as material. Table 3 summarises its material properties. Next to the panel geometry and material properties, load cases and constraints are required as input to the optimisation. The five load cases defined in tab. 4 determine in combination with the constraints (summarised in tab. 5) the feasible domain of the optimisation.

The layup retrieval tests include two sets of LPs. One is the set that Viquerat [8] uses for stacking sequence recovery. The other, generated for this paper, is the result of an optimised CFRP panel. Table 6 provides both LP sets. Viquerat’s LP set is not based on actual stackings or optimisation results. The chosen parameters are all close to zero. This way the LP set is placed close to the centre of the LP space feasible domain and chances for recovering stacking sequences are high. While the LP set of Viquerat represents an asymmetric unbalanced laminate, the optimisation result represents an almost symmetric balanced laminate close to the boundary of the feasible domain. The two extreme sets in combination allow a representative test of the layup retrieval algorithm.

The layup retrieval algorithm introduced in section 4 returns multiple stacking sequence solutions for a given set of LPs. Table 7 shows layup solutions for

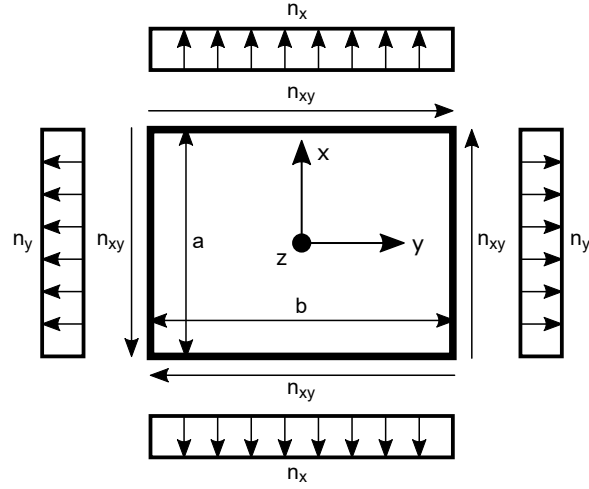


Figure 9: Simply supported benchmark panel ($a=1$ m, $b=1$ m)

Material	T-300 15k/976 @ 22 °C ($t_{ply} = 0.125$ mm)	
Stiffness		
E_1	133.70 GPa	
E_2	9.24 GPa	
G_{12}, G_{13}	6.27 GPa	
G_{23}	3.50 GPa	
ν	0.318	
Strength	Tension	Compression
σ_1	1427.21 MPa	1503.06 MPa
σ_2	39.02 MPa	206.84 MPa
τ_{12}, τ_{23}	76.53 MPa	

Table 3: Material definition for the benchmark panel[18]

Load case	n_x N m ⁻¹	n_y N m ⁻¹	n_{xy} N m ⁻¹
1	500 000	200 000	0
2	-50 000	-50 000	0
3	0	0	1000
4	100 000	0	10 000
5	-1000	0	10 000

Table 4: Load cases for the benchmark optimisation

Constraint type	Constraint	Value	Source
laminate design rules	allowed angles	0°, ±45°, 90°	[17]
	symmetric	Yes	[17]
	balanced	Yes	[17]
strength	tension	Tsai-Wu	[19]
	compression	Tsai-Wu	[19]
	shear	Tsai-Wu	[19]
stability	compression	HSB 45111-08	[20]
	shear	HSB 45112-02	[20]

Table 5: Summary of all optimisation constraints

Viquerat	1	2	3	4
V_x^A	0.2000	-0.0500	-0.1500	-0.1000
V_x^B	0.4000	0.2000	0.4000	0.2500
V_x^D	0.2000	0.2000	-0.0500	-0.1000
Optimisation	1	2	3	4
V_x^A	0.1784	0.0112	-0.7140	0.0213
V_x^B	0.0004	0.0000	0.0036	0.0013
V_x^D	0.0226	0.1164	-1.0000	0.0000

Table 6: Benchmark LP sets from Viquerat / optimisation

Sol.	θ_n												RMSE
	1	2	3	4	5	6	7	8	9	10	11	12	
0	69.8	-45.1	42.3	-36.6	-40.5	-82.4	-4.7	-33.8	24.6	54.7	-2.1	12.6	$1.73 \cdot 10^{-8}$
1	68.6	-53.2	43.6	-29.9	-29.2	-43.2	-77.4	-4.9	50.9	6.9	32.0	-3.5	$2.38 \cdot 10^{-8}$
2	69.0	-49.4	43.7	-22.9	-57.3	-39.5	-26.0	76.8	12.2	14.4	36.6	-5.3	$3.50 \cdot 10^{-8}$
3	69.1	-50.8	44.1	-21.9	-44.6	-54.5	-23.6	74.6	6.8	17.6	35.8	-5.4	$1.76 \cdot 10^{-8}$
4	70.5	-47.3	43.7	-26.5	-41.0	-70.7	-30.0	21.0	63.3	1.8	-7.4	25.0	$4.07 \cdot 10^{-8}$
5	61.3	-31.2	-60.6	56.2	-61.2	-27.9	15.7	-35.6	-8.4	61.2	18.6	9.0	$2.47 \cdot 10^{-8}$
6	55.4	-44.0	85.6	-29.5	-42.4	35.4	-55.9	-1.2	-16.4	58.4	19.3	6.1	$3.21 \cdot 10^{-8}$
7	73.8	-34.4	42.8	-54.3	-43.7	-34.8	32.5	83.0	-9.4	-9.1	7.7	32.6	$1.52 \cdot 10^{-8}$
8	52.8	-52.8	76.8	-26.6	-40.6	-21.7	-60.1	30.7	12.3	57.2	-8.6	11.3	$2.43 \cdot 10^{-8}$
9	61.1	-36.8	-70.8	47.8	-30.3	-29.6	-63.1	-8.4	23.4	56.2	-5.0	14.6	$2.14 \cdot 10^{-8}$

Table 7: Viquerat’s layup solutions in the 12 layer space, 500 iterations, $\text{RMSE}_{max} = 1e-7$, first solution 2.5s, all solutions 77.2s

Sol.	θ_n												RMSE
	1	2	3	4	5	6	7	8	9	10	11	12	
0	44.5	-43.8	45.1	-46.4	-40.6	15.1	-10.8	38.0	47.2	-44.9	-44.6	44.4	$8.39 \cdot 10^{-3}$
1	44.5	-43.9	45.1	-46.3	-40.7	15.1	-10.7	38.0	47.2	-44.9	-44.6	44.4	$8.39 \cdot 10^{-3}$
2	44.7	-44.7	-44.5	44.8	40.9	-10.6	13.8	-46.8	-40.2	44.6	-45.0	44.6	$8.67 \cdot 10^{-3}$
3	44.7	-43.9	44.4	-45.5	-41.9	14.1	-10.0	45.3	40.1	-44.8	-44.6	45.4	$8.50 \cdot 10^{-3}$
4	44.7	-44.6	-45.0	45.4	39.7	-11.1	14.6	-39.4	-48.1	44.7	-43.4	44.6	$8.62 \cdot 10^{-3}$
5	44.7	-44.7	-44.5	45.1	40.6	-10.7	13.9	-46.8	-40.2	44.6	-45.0	44.6	$8.67 \cdot 10^{-3}$
6	44.7	-44.6	-45.0	45.2	40.1	-11.0	14.5	-39.6	-47.9	44.6	-43.5	44.6	$8.62 \cdot 10^{-3}$
7	44.5	-43.8	45.0	-46.5	-40.5	15.1	-10.8	38.0	47.2	-44.9	-44.6	44.4	$8.40 \cdot 10^{-3}$
8	44.7	-44.3	44.3	-43.8	-43.5	13.9	-9.8	45.4	40.2	-44.7	-44.6	45.3	$8.50 \cdot 10^{-3}$
9	44.7	-44.7	-44.5	44.9	40.9	-10.6	13.8	-46.8	-40.3	44.6	-45.0	44.6	$8.67 \cdot 10^{-3}$
10	44.7	-44.6	-45.0	45.4	39.8	-11.0	14.6	-39.5	-48.0	44.6	-43.5	44.6	$8.62 \cdot 10^{-3}$

Table 8: Optimisation layup solutions in the 12 layer space, 500 iterations, $\text{RMSE}_{max} = 1e-2$, first solution 6.0s, all solutions 70.2s

Viquerat’s LP set⁶. They were obtained in 500 iterations using a $\text{RMSE}_{max} = 1e-7$. The calculation took 77.2 seconds with the first solution being available after 2.5 seconds. Table 8 shows layup solutions for the optimised LP set. Here 11 solutions were generated in 500 iterations with a $\text{RMSE}_{max} = 1e-2$. The first solution was available after 6.0 seconds while the entire calculation took 70.2 seconds.

The stacking sequence retrieval algorithm introduced in this paper allows a rapid transformation of LP sets into laminate layups. The retrieval time is hereby independent of the given LP set. Precise solutions ($\text{RMSE} \leq 1e-7$) are possible for evenly distributed laminates. Such laminates usually have a broad variety of solutions. Less accurate ($\text{RMSE} \leq 1e-2$), but still applicable solutions are found for more extreme laminates⁷. In all cases multiple solutions

⁶All layup solutions presented in this section are rounded to one decimal.

⁷It is worth mentioning that even though no manufacturing constraints are used during layup retrieval, the optimised LP set seems to include the layer angle limitations.

are available on an office notebook in less than one minute. The desired accuracy determines the number of available solutions.

The layer number can be selected freely in the layup retrieval algorithm. Usually the layer number is calculated from the ply thickness, depending on the material, and the total thickness of the laminate. The optimised CFRP panel has a total thickness of $t = 5.97$ mm. The ply thickness of T-300 15k/976 is given as $t = 0.125$ mm. This results in a 48 layer laminate. The layup retrieval algorithm finds 62 layup solutions in the 48 layer space within 158.4 seconds using 100 iterations. Setting the maximum RMSE to $\text{RMSE}_{max} = 1e-2$, the best layup solutions result in a $\text{RMSE} \leq 4e-3$. In the 12 layer space, the algorithm finds 11 layups in 500 iterations (success rate 2%). For 48 layers 100 iterations result in 62 layups (success rate 62%). This example shows how the number of available layup solutions significantly increases with the selected ply number.

Viquerat's method determines all 112 layup solutions for his LP set in 135 minutes⁸. Our layup retrieval algorithm finds the same number of solutions in 60000 (58675) iterations taking 158 minutes⁹. The accuracy is set to $\text{RMSE}_{max} = 1e-7$ in order to only obtain numerically exact results. Since Viquerat doesn't provide the full list of layup solutions, a comparison of the single laminates is not possible here. However, the layup retrieval algorithm finds the same number of valid and highly accurate solutions. Since Viquerat identifies all layup solutions and our algorithm finds the same number of valid solutions, we can assume that they are identical. This underlines the algorithm's capability to find all existing solutions in the 12 layer space. It is expected that the ability of the algorithm to find all solutions in the 12 layer space can be transferred to finding all solution spaces in higher layup spaces.

Liu's layup retrieval algorithm finds layup solutions for an arbitrary number of layers and a fairly restricted set of admissible ply angles [14]. The 12 layer solution of Liu's algorithm for the optimised LP set allowing 0° , $\pm 45^\circ$ and 90° layer angles is shown in table 9. The RMSE of Liu's solution is approximately twice the RMSE of the solutions presented in table 8. However, the advantage of Liu's solution is the strict limitation to the given set of admissible ply angles. A layup recovery of Viquerat's LP set is not possible with Liu's algorithm, because the solution time increases exponentially with the admissible layer angles. Also a solution in the 48 layer space is not feasible because of the increased calculation time¹⁰.

The introduced layup retrieval algorithm allows the stacking sequence recovery of arbitrary LP sets. Multiple layup solutions can be found on office notebooks in less than one minute. The desired accuracy has a strong influence on the number of layup solutions. For evenly distributed laminates, close to

⁸Using 16 threads on an Intel Xeon W-2145.

⁹Using 1 thread on an Intel Core i7-8758H.

¹⁰The attempt to retrieve a 48 layer laminate with 0° , $\pm 45^\circ$ and 90° layer angles on an office notebook was aborted after 24 hours

Sol.	θ_n												RMSE
	1	2	3	4	5	6	7	8	9	10	11	12	
0	45.0	-45.0	-45.0	45.0	45.0	0.0	0.0	-45.0	-45.0	45.0	-45.0	45.0	$1.97 \cdot 10^{-2}$

Table 9: The layup solution of Liu’s algorithm for the optimised LP set in the 12 layer space took 53.4 seconds

the centre of the LP feasible domain, numerically exact solutions are possible with a RMSE $\leq 1e-7$. Solutions for more extreme laminates are available with less accuracy ¹¹. However, the accuracy is still high compared to other available retrieval solutions. The solution accuracy increases with the number of plies. Also the success rate (fraction of solutions per iterations) dramatically increases with the number of plies. In engineering applications, where laminate thicknesses above 1.5 mm are common, both circumstances accelerate the layup retrieval.

6. Conclusion

This paper identifies the need for a universal stacking sequence retrieval algorithm to transform LP sets into laminate layups. A thorough analysis of the relationship between layup and LP space is the foundation for such an algorithm. An important finding in this context are the solution spaces embedded in each layup space. The existence of multiple, valid solution spaces prevents the successful application of global optimisation approaches. Gradient based approaches are generally more suitable for smooth problems, but only find a single optimum. However, a successful algorithm for the stacking sequence retrieval needs the ability to find all global optima.

The authors propose a multi start optimisation algorithm solving the layup retrieval problem. The layup retrieval algorithm allows subminute transformation of LP sets into layup solutions. Multiple layups are retrieved for arbitrary LP sets with high solution accuracy. Increasing the number of plies has a positive effect on the layup solution accuracy while also decreasing the solution time. The layup retrieval algorithm doesn’t take into account any manufacturing constraints. However, manufacturing constraints can be considered during optimisation by modifying the lamination parameter feasible domain. The great number of solutions for a single LP set further allows the selection of the most favorable laminates with regard to the given boundary conditions.

Future research is necessary for a further development of lamination parameter based composite optimisation. The algorithm presented in this paper allows a rapid transformation of LP sets into laminate layups. While the algorithm

¹¹The low accuracy for the extreme laminates possibly result from slightly invalid LP sets. Such LP sets can hardly be avoided in the optimisation and should be covered by a robust layup retrieval algorithm.

returns all existing solutions, the assessment and selection of one layup solution is not part of this publication. The retrieval algorithm does not take into account manufacturing constraints, but they can be considered in the selection of an adequate layup solution.

The present work suggests the existence of an infinite number of layup solutions for laminates with more than 12 layers. In contrast only a finite number of them exist in 12 layer space. Performing layup retrieval in 12 layer space thus allows finding all possible solutions. Layer-and-angle redundancy permits upscaling of 12 layer solutions to higher layer spaces. In higher layer spaces each single solution expands to a solution space. A solution space allows choosing a layup solution considering additional constraints, such as for manufacturing.

Declaration of Competing Interest

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

CRedit

Moritz Sprengholz: Methodology, Software, Formal analysis, Investigation, Validation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization **Hendrik Traub:** Conceptualization, Software, Formal analysis, Investigation, Validation, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration **Michael Sinapius:** Writing - Review & Editing, Supervision, Project administration, Funding acquisition **Sascha Dähne:** Software, Resources **Christian Hühne:** Writing - Review & Editing, Supervision, Project administration, Funding acquisition

Acknowledgments

We would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanys Excellence Strategy – EXC 2163/1- Sustainable and Energy Efficient Aviation – Project-ID 390881007.

Data availability

The processed data required to reproduce these findings are available to download from <https://git.rz.tu-bs.de/m.sprengholz/publication-lamination-parameters/>.

References

- [1] Y. Xu, J. Zhu, Z. Wu, Y. Cao, Y. Zhao, W. Zhang, A review on the design of laminated composite structures: Constant and variable stiffness design and topology optimization, *Advanced Composites and Hybrid Materials* 1 (3) (2018) 460–477. doi:[10.1007/s42114-018-0032-7](https://doi.org/10.1007/s42114-018-0032-7).
- [2] S. Nikbakt, S. Kamarian, M. Shakeri, A review on optimization of composite structures Part I: Laminated composites, *Composite Structures* 195 (2018) 158 – 185. doi:<https://doi.org/10.1016/j.compstruct.2018.03.063>.
- [3] S. Setoodeh, M. Abdalla, Z. Gurdal, *Approximate Feasible Regions for Lamination Parameters*, 2006. doi:[10.2514/6.2006-6973](https://doi.org/10.2514/6.2006-6973).
- [4] V. B. Hammer, M. P. Bendsøe, R. Lipton, P. Pedersen, Parametrization in laminate design for optimal compliance, *International Journal of Solids and Structures* 34 (4) (1997) 415–434. doi:[10.1016/S0020-7683\(96\)00023-6](https://doi.org/10.1016/S0020-7683(96)00023-6).
- [5] M. Picchi Scardaoni, M. Montemurro, Convex or non-convex? on the nature of the feasible domain of laminates, *European Journal of Mechanics - A/Solids* 85 (09 2020). doi:[10.1016/j.euromechsol.2020.104112](https://doi.org/10.1016/j.euromechsol.2020.104112).
- [6] M. A. Albazzan, R. Harik, B. F. Tatting, Z. Gürdal, Efficient design optimization of nonconventional laminated composites using lamination parameters: A state of the art, *Composite Structures* 209 (2019) 362–374. doi:[10.1016/j.compstruct.2018.10.095](https://doi.org/10.1016/j.compstruct.2018.10.095).
- [7] M. Bloomfield, J. Herencia, P. Weaver, *Optimisation of Anisotropic Composite Plates Incorporating Non-Conventional Ply Orientations*, 2008. doi:[10.2514/6.2008-1918](https://doi.org/10.2514/6.2008-1918).
- [8] A. D. Viquerat, A continuation-based method for finding laminated composite stacking sequences, *Composite Structures* 238 (2020) 111872. doi:[10.1016/j.compstruct.2020.111872](https://doi.org/10.1016/j.compstruct.2020.111872).
- [9] X. Liu, C. Featherston, D. Kennedy, Two-level layup optimization of composite laminate using lamination parameters, *Composite Structures* 211 (Dec. 2018). doi:[10.1016/j.compstruct.2018.12.054](https://doi.org/10.1016/j.compstruct.2018.12.054).
- [10] R. M. Jones, *Mechanics of Composite Materials*, CRC Press, 1999. doi:[10.1201/9781498711067](https://doi.org/10.1201/9781498711067).
- [11] S. W. Tsai, N. J. Pagano, *Invariant properties of composite materials*, Tech. rep. (Jan. 1968).
- [12] H. T. Hahn, S. W. Tsai, *Introduction to Composite Materials*, CRC Press, 1980.

- [13] T. Macquart, N. Werter, R. De Breuker, Aeroelastic Design of Blended Composite Structures Using Lamination Parameters, *Journal of Aircraft* 54 (2) (2017) 561–571. doi:[10.2514/1.C033859](https://doi.org/10.2514/1.C033859).
- [14] X. Liu, C. A. Featherston, D. Kennedy, Two-level layup optimization of composite laminate using lamination parameters, *Composite Structures* 211 (2019) 337–350. doi:[10.1016/j.compstruct.2018.12.054](https://doi.org/10.1016/j.compstruct.2018.12.054).
- [15] C. Cartis, J. Fiala, B. Marteau, L. Roberts, Improving the Flexibility and Robustness of Model-based Derivative-free Optimization Solvers, *ACM Transactions on Mathematical Software* 45 (3) (2019) 1–41. doi:[10.1145/3338517](https://doi.org/10.1145/3338517).
- [16] H. Beyer, B. Sendhoff, Simplify your covariance matrix adaptation evolution strategy, *IEEE Transactions on Evolutionary Computation* 21 (5) (2017) 746–759. doi:[10.1109/TEVC.2017.2680320](https://doi.org/10.1109/TEVC.2017.2680320).
- [17] E. Werthen, S. Dähne, Design- and manufacturing constraints within the gradient based optimization of a composite aircraft wingbox, in: 6th Airframe Structural Design Conference, Bristol October 2018, 2018.
- [18] U. Dept Of Defense, Composite materials handbook-MIL 17: Volume 2: Polymer matrix composites: Materials properties, *Composite Materials Handbook-MIL 17: Volume 2: Polymer Matrix Composites: Materials Properties* (Jan. 2000).
- [19] A. Khani, S. T. Ijsselmuiden, M. M. Abdalla, Z. Gürdal, Design of variable stiffness panels for maximum strength using lamination parameters, *Composites Part B: Engineering* 42 (3) (2011) 546–552. doi:[10.1016/j.compositesb.2010.11.005](https://doi.org/10.1016/j.compositesb.2010.11.005).
- [20] *Luftfahrttechnisches Handbuch (LTH): Handbuch Strukturberechnung (HSB)*, issue c Edition, IASB, München, 2009.