An Approach Towards an Optimal Design of Composite Structures Using Abaqus as FE-solver

M. Prackwieser[a], M. Gratt[a] and B. Goller[a*]

[a]INTALES GmbH Engineering Solutions, Innsbrucker Str. 1, 6161 Natters, Austria
*goller@intales.com

Abstract: *The present paper shows the development of an efficient, fast and reliable optimizer for composite and metallic parts of lightweight structures. The algorithm aims at identifying the optimal configuration of different structural parts concerning thickness, fibre orientation, number of plies, etc. This leads to mass savings and also a decrease of the development time in the structural dimensioning phase. Due to the limited applicability of classical optimization algorithms like gradient based or evolutionary methods in case of large Finite Element models with a high number of design variables, a novel approach is presented where the optimization problem is tackled by a heuristic adaption procedure on element level. This approach will be illustrated on small numerical examples to show the functionality of our optimizer. Abaqus user subroutines provide accessible interfaces which are embedded in our optimization workflow in order to apply the optimization routine to large industrial problems. An example for such a problem is given at the end of this paper to demonstrate the smooth interaction between optimization routine and Abaqus.*

Keywords: *Aircraft, Composites, Design Optimization, Minimum-Weight Structures, Optimization, Output Database, Post-processing, Shell Structures.*

1. Introduction

One main aspect for the competitiveness in the design of lightweight structures is the identification of weight saving opportunities while fulfilling multiple constraints in terms of allowable values for stresses, strains, displacements etc. The variables to be optimized are the number of plies of different parts, the orientation of the fibres and also their stacking sequence in case of composite parts. For metallic parts, the thicknesses are to be optimized, while the diameter represents the design variable related to fasteners. For real structures, the number of variables to be optimized might soon reach a limit where classical optimization algorithms, like e.g. gradient-based methods or also evolutionary algorithms, might not be applicable due to the curse of dimensionality. An infeasible high number of model evaluations would be required in order to identify a possible solution.

The topic of the here described project is the development of a novel approach where the optimization problem is tackled by a heuristic adaption process on the element level. More specifically, the starting point of the algorithm is the element stiffness matrix, which is adjusted according to the temporary state in terms of failure criteria at a certain load level. The external forces are applied in several load increments until the full load level is active and during each of these increments an adaptation process is performed. The process can be compared with the mechanical adjustment of biological materials, like wood and bones, where the stiffness formation process is determined by the acting forces.

The key aspect in this process is the access of the element routines and the development of an adaptation strategy for each element type in order to obtain optimal convergence. A number of state-of-the-art element formulations have therefore been implemented and linked with the commercial Finite-Element (FE) code Abaqus. In order to manage the independently adjusted properties of each element, an efficient data-management is required which has been solved by connecting an SQLITE-database to the element routines, where the values of each iteration of all variables to be adjusted are written and accessed in the next iteration. Strategies for the type and magnitude of change of the distinct design variables have been investigated where especially the robustness of the solution, i.e. its sensitivity with respect to small deviations of input parameters, is in the focus of the studies.

The main advantage of this approach is its applicability to large and complex FE-models, where also several million degrees of freedom do not lead to algorithmic limitations. In addition, the strategy can be used for models involving different kinds of elements, like shell elements including layered composite, sandwich or isotropic (metallic) shell sections, and also fastener elements.

The structure of the present paper is as follows: after a brief discussion of the theoretical background in Sec. 2, the implementation of the element routines in the framework of Abaqus with the link to the SQLITE-database is addressed in Sec. 3. The augmentation of the element routines by failure criteria and the respective strategies for adaptation are topic of Sec. 4. The quality of the employed adaptation strategies is judged by means of some small test examples.

2. Theoretical background

The goal of the present project is the optimal design of lightweight structures such that the weight is minimized and all design criteria are fulfilled. This can be expressed as a constrained optimization problem, i.e.

$$\min \; f(x) \tag{1}$$

$$\text{subject to } \; g_i(x) \le c_i \;\text{ for }\; i = 1 \ldots m, \tag{2}$$

where x are the design variables, like thickness, fibre angle, fastener type, etc., $f(x)$ is the objective function, i.e. the total mass, and $g_i(x)$ are the constraints which are defined by failure modes, design rules and feasibility of construction. Classical optimization algorithms might soon reach their feasibility limits since the number of design variables is in the range of hundreds or even thousands for large FE-models.

Hence, an alternative approach based on an iterative adaption theory on element level is followed in this paper, which is inspired by ideas like "fibre steering" (aligning the fibres along the most effective direction), "computer aided internal optimization" (Kriechbaum, 1992), and "multidomain topology optimization" (Ma, 2006). More specifically, during the structural analysis the status of each element in terms of fulfilling the investigated criteria is checked and - if necessary - the values of the design variables of the element are changed. This change may imply an increase of the thickness in case of a metal, the change of the draping angle or of the stacking sequence in case of composite elements or also the increase of the diameter in case of a fastener element. In order to consider a possible load distribution when changing the stiffness of the structure due to this local adjustment, the external load is applied iteratively. In this way, the structure can "grow" according to the acting load, which is a concept inspired by mechanical adaptation of biological materials. As it also applies to wood, the growth is only additive and hence, the material is never reduced. In this way, it can be ensured that in case of multiple load cases only the critical load cases are driving for each element, while a lower external force does not change the element properties.

The crucial point is the definition of the adaptation strategy, which has to be defined for each failure criterion. A measure of the degree to which the limit factor is exceeded for each of the criteria is the so-called reserve factor (RF), which defines the factor by which the load is increased or decreased, respectively, in order to reach the limit state given by an RF equal to 1.0. Hence, this value is a measure by which factor e.g. the thickness of a metallic element or the diameter of a fastener is to be increased. In case of multiple design variables belonging to one element or to one criterion, the strategy becomes more involved and has to be specific to each case. The approaches which have been developed will be described in detail in the following section.

3. Development of the optimization procedure

The idea behind the optimization process is the adaptation of the properties of each element according to its state with respect to the failure criteria. Hence, for each material/element type a strategy for adjusting the element has been developed and implemented in the element routine (see Sec. 5.1). Based on the current values of the element forces, moments and/or stresses, the design variables of the analysed element are adjusted.

In the following, the strategies applied to the different element types are discussed and an example will illustrate the functionality of the algorithm.

3.1 Composite elements

The first kind of failure criteria which is investigated are composite failure modes. To quantify an eventual failure, we calculate for each criterion the quotient of actual occurring value of strain or stress and the corresponding threshold marking failure. Thus we obtain a degree of utilization. A value above 1 means failure.

One frequently applied approach is the maximum strain criterion. The requirement is formulated such that the maximum strain ε_1 occurring in an element must not exceed a user-defined threshold, which is usually in the range of 0.3% to 0.4%:

$$\text{Maximum strain criterion}:\begin{cases} f=\dfrac{\varepsilon_1}{\varepsilon_t} \text{ for } \varepsilon_1 \geq 0 \\ f=\dfrac{\varepsilon_1}{\varepsilon_c} \text{ for } \varepsilon_1 < 0 \end{cases} \tag{3}$$

$\varepsilon_t \ldots$ allowable tension strain

$\varepsilon_c \ldots$ allowable compression strain

In addition, two other criteria are applied, which are based on comparison stresses. In case of fibre failure the Yamada-Sun criterion is relevant because it compares the acting and allowable stresses in fibre direction:

$$\text{Yamada} - \text{Sun criterion}:\begin{cases} f=\sqrt{\left(\dfrac{\sigma_1}{X_t}\right)^2+\left(\dfrac{\sigma_{12}}{S}\right)^2} \text{ for } \sigma_1 \geq 0 \\ f=\sqrt{\left(\dfrac{\sigma_1}{X_c}\right)^2+\left(\dfrac{\sigma_{12}}{S}\right)^2} \text{ for } \sigma_1 < 0 \end{cases} \tag{4}$$

$X_t \ldots$ allowable tension stress in fibre direction

$X_c \ldots$ allowable compression stress in fibre direction

$S \ldots$ allowable shear stress

If the Hashin criterion is the decisive criterion we can assume that the matrix fails, because the stresses perpendicular to the fibre direction have been exceeded:

$$\text{Hashin criterion}:\begin{cases} f=\sqrt{\left(\dfrac{\sigma_2}{X_t}\right)^2+\left(\dfrac{\sigma_{12}}{S}\right)^2} \text{ for } \sigma_2 \geq 0 \\ f=\sqrt{\left(\dfrac{\sigma_2}{X_c}\right)^2+\left(\dfrac{\sigma_{12}}{S}\right)^2} \text{ for } \sigma_2 < 0 \end{cases} \tag{5}$$

$X_t \ldots$ allowable tension stress in fibre direction

$X_c \ldots$ allowable compression stress in fibre direction

$S \ldots$ allowable shear stress

Thus we obtain three degrees of exploitation in which the highest value is corresponding to the maximum material utilization.

The following strategy adapts the layup to meet all criteria. It can be applied to both four-node quadrilateral and three-node triangular shell element. In addition, also for the dimensioning of the face sheets and core thickness of sandwich elements and of the curved element this approach is employed.

3.1.1 Strategy to adapt the composite structure

In case the requirement is not fulfilled, the following adaptation strategy is applied: the section forces for initially isotropic material properties (smeared stiffness of an initial stacking) are calculated at each element centre. The load of all load cases is applied in several increments and thus we obtain the first section forces for an incremental load of e.g. 10%. Then the optimization routine starts to search the dominating direction for each element using a heuristic method. The result is the draping angle of the stacking which suits the most with the applied loads and generates therefore the lowest degree of utilization for the element. Further, a grouping parameter *g* is calculated, which depends on the degree of anisotropy of the stress state. The proposed stacking then follows a strict sequence, $45/90/\text{-}45/0_g$, which is repeated until the required membrane and bending stiffness is obtained. The parameter *g* describes how many plies with a direction of the draping angle follow each other. If these adjustments are not sufficient to fulfil all criteria, an additional ply is added to the stacking of the element and the entire structure is realigned in the following step. This is repeated until all load increments of all load cases are applied and a final stacking is obtained for all elements.

The approach will be illustrated using quadrilateral monolithic elements. For triangular elements or all other previous mentioned elements, the same strategy can be applied.

3.1.2 Numerical example: plate with hole

The first example used for illustrating the method is a plate with hole under tension load as it can be seen in Figure 1. The structure is clamped at the left end and loaded with constant point loads of 2000 N in positive x-direction at the right edge. The plate is made out of composite (fabric), where the initial configuration is given with 4 plies (45/0/0/45) with a ply thickness of 0.28 mm. The Young's modulus in both directions is given with 62000 N/mm^2.

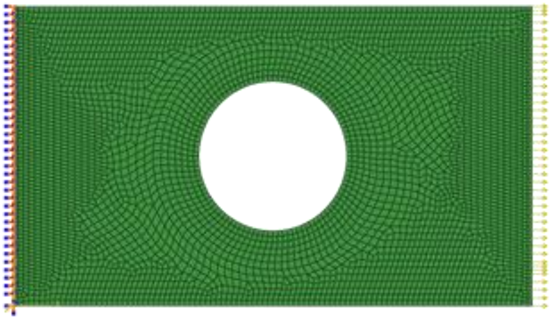


Figure 1. FE-model of the plate with hole under tension

The design variables are the number of plies and the draping angle, where the values after the optimization process are shown in Figures 2 and 3. The direction of the fibres is optimal if it is aligned with the direction of the main load path. Since a plate with hole under tension is a well-known benchmark example, the direction of the principal stresses is known from analytical solutions. This direction can be recognized also in Figure 2, where however, the curvature of the directions around the hole is not as pronounced as in the analytical solutions. This can be explained by the fact that the material is not isotropic due to the fibres and hence the optimal direction is not equal to the draping angle. The draping angles of the single elements as indicated in Figure 2 represent therefore a plausible solution.

The number of plies is shown in Figure 3 for each element of the model. As it is known from analytical solutions, the highest stresses are located at the top and bottom of the hole. These are also the two regions where the highest thickness increases are performed during the optimization process. The highest value for the number of plies has been set to 50 (user input prior to the analysis) and this is also the value which has been reached by the elements in these areas. The definition of a maximum values for the plies is motivated by the situation that an increase of the number of plies provokes a higher attraction of loads to this area (due to the higher stiffness) and hence an unrealistic thickening would occur around stress peaks. By limiting the number of plies to a user defined value, a certain re-distribution of the stresses around the location of the peak values can be obtained. The increase of the number of plies in the two regions from the hole to the support reflects the direction of the forces in the structure. Also at the support a certain increase can be seen due to the higher stresses.

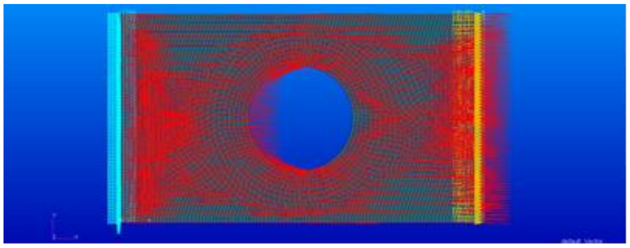


Figure 2. Draping angle after optimization

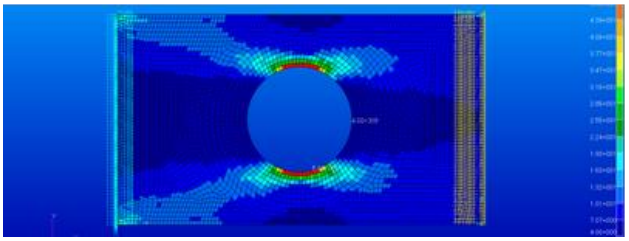


Figure 3. Number of plies after optimization

3.2 Metallic elements

In addition, a simultaneous optimization of the thickness of metallic parts is envisioned. Thus, the optimizer allows the structural dimensioning of hybrid parts, since the stiffnesses of metal and composite components are interdependent in terms of the global stiffness of the whole structure. The strategy of optimising is analogous to the composite one and is also based on a heuristic approach. The thickness of each shell-element is therefore adapted according to its state with respect to the von Mises yield criterion (calculated for a general plane stress load scenario)

$$\sigma_v = \sqrt{\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2 + 3\sigma_{12}^2} \qquad \frac{\sigma_v}{\sigma_{v,allowed}} \le 1 \tag{6}$$

$\sigma_{1/2} \cdots$ normal stresses

$\sigma_{12} \cdots$ shear stresses

$\sigma_v \cdots$ von Mises stresses

3.3 Fastener elements

To meet all requirements of an optimizer which should be able to support an engineer in the dimensioning of complex structures, also fasteners are considered in the optimising process. Three criteria assess an eventual failure of a fastener and thus steer its dimensioning. We take into account a combined failure criterion

$$\left(\frac{\sigma_n}{\sigma_{n,allowed}}\right)^2 + \left(\frac{\sigma_s}{\sigma_{s,allowed}}\right)^2 \le 1, \tag{7}$$

$\sigma_n \cdots$ acting and allowable normal stresses

$\sigma_s \cdots$ acting and allowable shear stresses,

a pull-trough failure criterion

$$\sigma_{PT} = \frac{P}{\frac{1}{4}(D-d)^2\pi} \quad \text{and} \quad \frac{\sigma_{PT}}{\sigma_{PT,allowed}} \le 1, \tag{8}$$

P $\ldots$ acting force
d $\ldots$ diameter of the collar
D $\ldots$ diameter of the head

and a bearing failure criterion, that compares the stresses which are transferred from the collar of the fastener to the plate material with the allowable bearing stresses

$$\sigma_{BEA} = \frac{P_{shear}}{d\pi t_0} \qquad \frac{\sigma_{BEA}}{\sigma_{BEA,allowed}} \le 1 \tag{9}$$

P_{shear} ... acting shear force
d ... diameter of the collar
t_0 ... thickness of thinnest connecting plate.

4. Smoothing to provide manufacturability

In the optimization procedure, manufacturing issues have to be considered meaning that the term "optimal configuration" of structural parts implies the condition of manufacturability. In terms of manufacturing issues especially a smooth surface is desired, meaning that there is no abrupt change in thickness or stacking of a part. Since in our approach each element can theoretically have a different stacking or thickness than its neighbour, it is necessary to smooth the element properties over the structure to get connected homogenous areas.

The idea is not to smooth the final stacking, but to pass the section forces already smoothed to the optimizing routine in every load increment. Since the structure is consequently adapted for the smoothed section forces, additional optimizing loops without increasing the load after the full load are performed in order to adapt the elements to the actually occurring section forces.

The smoothing has always to be understood as a compromise. We want to somehow maintain the information calculated from the optimization routine, but consider also the manufacturing process. Hence it seems reasonable to weight elements which are nearer to the considered element higher than others whose distance is larger. Thus we are going to use a weighting function φ which is dependent on the distance r between the centroids of two elements to determine the weights for the smoothing routine:

$$\varphi : r \to \exp\left\{-\frac{1}{2}\left(\frac{r}{\lambda}\right)^2\right\}, \tag{10}$$

where λ has to be chosen by the user (we set it to 1). The weighting φ is now strongly scale dependent. Therefore we have to transform the distances between neighbours from the interval $[0, R]$ to an interval $[0, b]$. The variable R is the radius within which the elements are considered as neighbours. We transform the intervals by the affine transformation:

$$\gamma_{[a,b],[c,d]} : x \to a + (b-a)\frac{x-c}{d-c}, \tag{11}$$

where we usually set $a = 0,\ b = 0,\ d = R.$ Now the weight vector is given by:

$$w_{ik}^{l} = w_{ik} = \varphi\Big(\gamma_{[a,b],[c,d]}(r_{ik})\Big), \text{ where } r_{ik} = \|x_i - x_k\|. \tag{12}$$

After all the weights are calculated the whole weight vector needs to be normalized:

$$\sum_{i=1}^{n_k} w_{ik} = 1 \text{ for all } k = 1,\dots,N. \tag{13}$$

Then, the smoothed section forces are given for every load increment l by

$$\widetilde{sf}_k^{\,l}(j) = \sum_{i \in N_k} w_i sf_k^l(j), \tag{14}$$

$$\widetilde{sm}_k^{\,l}(j) = \sum_{i \in N_k} w_i sm_k^l(j), \quad \text{for } k = 1,\dots,N \text{ and } j = 1,2,3, \tag{15}$$

where N_k is a set containing the indices of the elements belonging to the neighbourhood of the k-th element and N is the total number of elements of the model. The actually occurring section forces are expressed by sf_k and sm_k.

4.1 Numerical example for smoothing

We demonstrate the effects of smoothing on a cylindrical roof, which is loaded with a distributed load on the left half of the structure. Figure 4 shows the obtained number of plies after optimizing the roof with the strategy presented above. The blue surfaces are marking composite elements with few plies whereas the more the elements tend to appear red the thicker the stacking is.

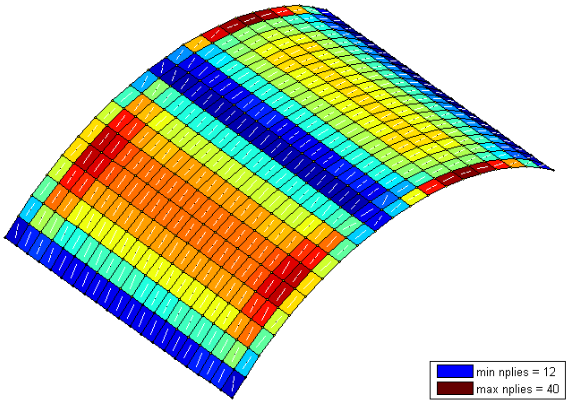


Figure 4. Number of plies after optimization

As first step we have to choose a suitable radius for the neighbour search. The radius R has a great influence to what extent the properties of the elements are smoothed. In Figure 5 we can see that all red elements affect the green in the middle. A smaller radius means to stay closer to the 'optimal' solution and a larger radius creates a more continuous stacking and thickness.

We set the radius for this example to R=50 and the angle for the allowed variation of the element normals to α=30° (Figure 5).

The result of the smoothing process is a roof with more homogenous areas (Figure 6). This is also evident in the maximum number of layers which are decreasing from 40 to 35. The total number of plies of the whole roof construction is increasing. Thus, the structure is less optimized with regard to the input data, i.e. it has a higher weight. However the advantage is that the gained result after smoothing is less sensitive to slight changes in the input parameters and is therefore more robust.

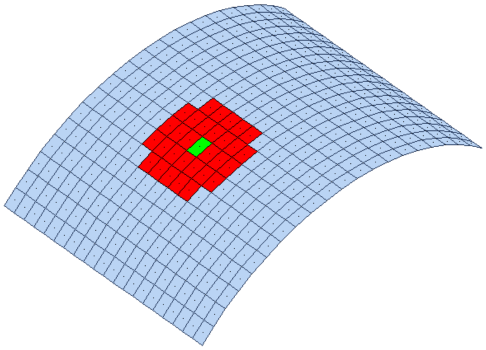


Figure 5. R-α-neighbourhood (red) of an element (green)

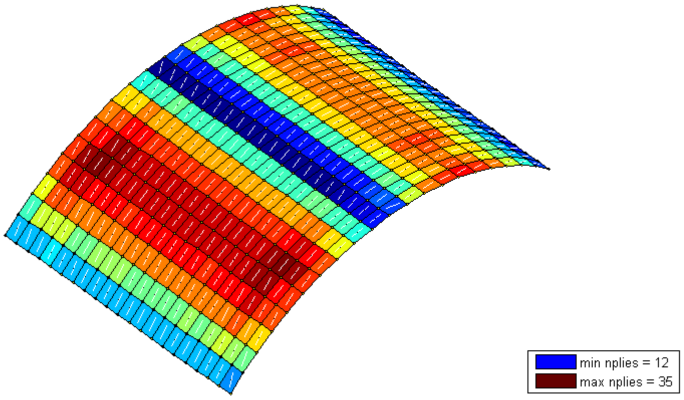


Figure 6. Number of plies after smoothing

5. Embedment of ABAQUS in the optimization process

In order to implement the proposed optimization algorithm at element level, an access to the element routine has to be provided. This is necessary since the layup of the shell elements and the diameter of the fasteners are changing between different steps of one Abaqus job. This means that the mechanical and geometrical properties, which are required for the definition of the element stiffness matrix, are changed according to the demands of the optimization process. The data to exchange the properties of the individual elements are handled over an external database. Every single element has to access this database before calculating its local stiffness for the global assembly. Therefore full access to the element routine is crucial to render the interventions of the optimizer possible. Abaqus provides interfaces for the user-defined interference in the solution process (Abaqus User Subroutines Reference Manual). The user subroutine employed for the implementation of the element routines is called **UEL**. Beside the mandatory committal of the element stiffness matrix to the Abaqus solver, several parameters, as for example the current step number, can be read in and out over this interface. This allows us to steer the optimizer via the information provided by the subroutine. Thus, in order to make the optimizer applicable to a certain region of a model, the type of element used for that component must be implemented as a user subroutine. Hence, the most frequently applied element types have been coded in FORTRAN in terms of the **UEL** for Abaqus, which are three- and four-node shell elements, a sandwich element, an element which can model the actual geometry of a curved part and a fastener element. All element formulations are state-of-the-art and were compared with success to the Abaqus elements to which they correspond.

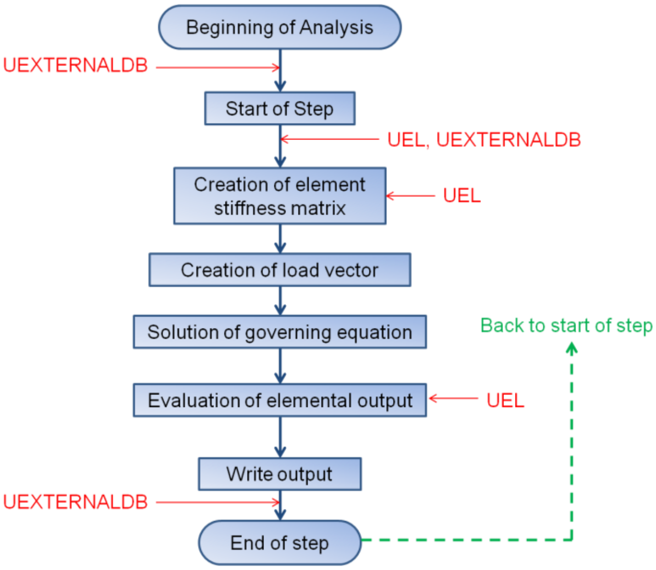
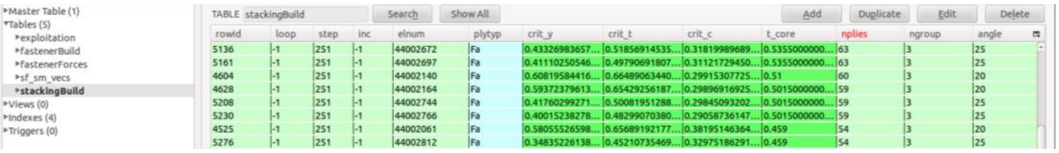


Figure 7. Solution sequence with interfaces to user-subroutines

Figure 7 shows the sequence of a linear static analysis in Abaqus and those stages of the analysis where the user subroutines needed by the optimization algorithm are invoked.

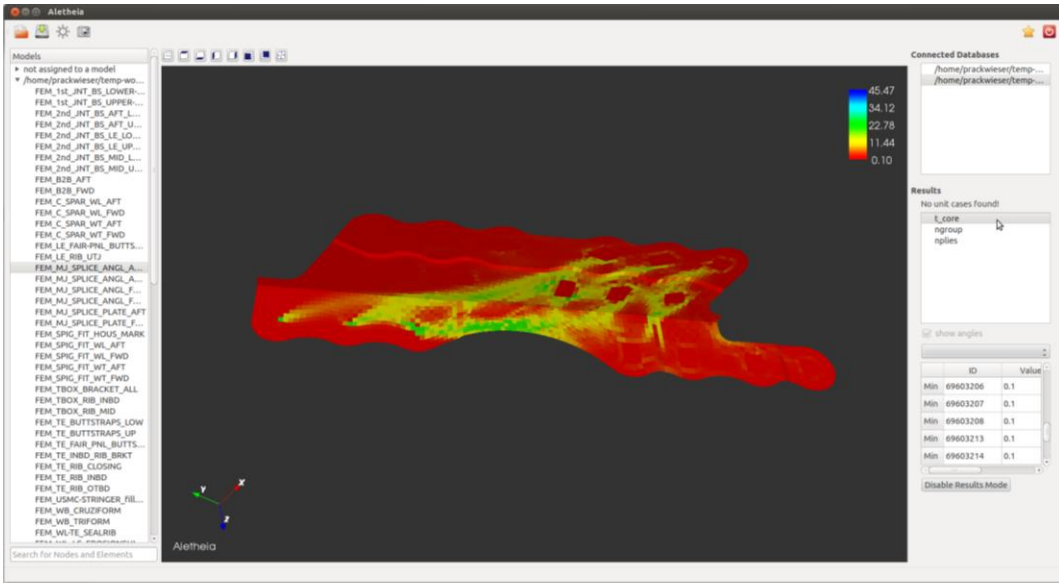
As already mentioned, we need to manage the results of the optimization process over a database in order to provide them for the consecutive calculation. The user subroutine **UEXTERNALDB** provides an interface between Abaqus and an external database. For complex Finite Element models, a large amount of data has to be handled since the current properties of each element have to be stored in every iteration and for each load case and retrieved in the next iteration. An SQLITE-database provides a suitable means for this problem since data can be stored and accessed efficiently. The optimization algorithm is based on adaption of the element properties during an iteratively increased external load. In order to handle this iterative increase of the load while performing a full analysis as shown in **Error! Reference source not found.**7, a step in Abaqus is understood as an iteration of the optimizer for one load case. For this purpose multiple steps are defined in the Abaqus input file. For a pre-defined number of *n* iterations, *n* steps have to be defined for each load case *k*, where in the *i*-th step the total external load is multiplied with the factor *i/n*. All together are resulting *n* times *k* steps for an optimization run. The optimizer creates a separate database for every load iteration and load case which contains the current layup and the exploitation of each element according to failure criteria from Sec. 3. Thus, the full history of the optimization process is available. After all load cases were calculated for a load increment, a database is created which summarizes all results for this load increment and in which the different stacking per element are merged to a new basis for the next load increment. Thereby, the results of the optimization run can be traced completely.



rowid	loop	step	inc	elnum	plytyp	crit_y	crit_t	crit_c	t_core	nplies	ngroup	angle
5136	-1	251	-1	44002672	Fa	0.43326983657...	0.51856914535...	0.31819989689...	0.5355000000...	63	3	25
5161	-1	251	-1	44002697	Fa	0.41110250546...	0.49790691807...	0.31121729450...	0.5355000000...	63	3	25
4604	-1	251	-1	44002140	Fa	0.60819584416...	0.66489063440...	0.29915307725...	0.51	60	3	20
4628	-1	251	-1	44002164	Fa	0.59372379613...	0.65429256187...	0.29896916925...	0.5015000000...	59	3	20
5208	-1	251	-1	44002744	Fa	0.41760299271...	0.50081951288...	0.29845093202...	0.5015000000...	59	3	25
5230	-1	251	-1	44002766	Fa	0.40015238278...	0.48299070380...	0.29058736147...	0.5015000000...	59	3	25
4525	-1	251	-1	44002061	Fa	0.58055526598...	0.65689192177...	0.38195146364...	0.459	54	3	20
5276	-1	251	-1	44002812	Fa	0.34835226138...	0.45210735469...	0.32975186291...	0.459	54	3	25

Figure 8. Screenshot of the results database (each row for an element)

Although the results in the SQLITE-database format (Figure 8) are easy to handle, a graphical illustration of the results becomes necessary for a quick overview or to place the results at the disposal for customers or other potential users. User elements cannot be displayed in Abaqus /CAE and the results for section forces and stresses are only available in the database. Therefore, we developed our own viewer (called ALETHIA) to provide the visualization of the results.



No unit cases found!

	ID	Value
Min	69603206	0.1
Min	69603207	0.1
Min	69603208	0.1
Min	69603213	0.1
Min	69603214	0.1

Figure 9. Aletheia viewer to illustrate results by accessing the SQLITE database

6. Industrial example

Finally, the optimizer is applied to a large industrial example. The level of detail is decisive here, since even small components can cause several thousand parameters to optimize in case of a configuration with multiple different composite layup areas consisting of several materials with several failure criteria and consequently with different constraints for each element. This creates an optimization task with a vast multi-dimensional parameter space, several constraints depending on the properties of the element and the objective to find a configuration of the structure which is able to carry the applied load cases and whose parts are not oversized at the same time, meaning that the result is as light as possible.

The structure we consider in the following consists of metal parts and composite components which in turn can be divided according to their setup into tape, fabric and sandwich areas. Several different bolt types serve as fastener to connect the different parts. Thus, all components are included which can be part of a lightweight structure.

The FE-model we consider consists of over 800.000 elements, thereof 460.000 shell elements. These in turn can be assigned to 2000 different stacking and setups in the start configuration. Summing up, the whole model is defined by over four million degrees of freedom.

Depending on the configuration of the FE-simulation, the time which is necessary for one step in Abaqus (linear computation to converge for a certain load level) varies from less than an hour to several hours when contact between different parts of the model is fully considered (over contact pairs). The functioning of the optimizer causes that this time has to be multiplied by about 11: 10 times to apply the load with an increment of 10% and a last time to calculate the exploitation of the final structure (see Section 4). Since the 45 load cases are independent, they have to be applied successively, and as such the total amount of steps which has to be evaluated rises to 495. The high amount of steps in one Abaqus job and the consideration of contact in the calculation lead to two problems for which workarounds were necessary.

The first problem is to define two user defined elements as contact pair which is not possible by default. The simplest solution we found is to cover all user elements with "dummy" elements. In doing so, we assign to Abaqus built-in elements (for example S4 or S3) a negligible stiffness and no weight so that they are not affecting the calculation. We use these elements then to define master and slave of the contact pair.

The second problem is the number of `CLOADS` per step. The load is defined over an include file for each load case which consists of about two million lines with node, degree of freedom and value. By starting a job with more than 35 steps the pre.exe of Abaqus terminates with an ERROR message because unit 25 exceeds the limit of 16 GB. Since this limit cannot be increased currently the only solution was to divide the job into several jobs with restarts and adapt our FORTRAN tools to this new circumstance. Python tools are generating the input-files for the start and the restarts and a bash file guarantees that the whole optimization process is still running automatically.

Once the workarounds are implemented, the optimizer is applicable to the presented and any other lightweight structure regardless of the size. Since the 45 load cases are acting independent, each can be seen as individual optimization task. In Figure 10 and Figure 11 the resulting number of plies and thicknesses of two components out of the full structure are shown. It shall be noted that for this analysis no smoothing procedure has been applied, the results hence represent for each element the optimal configuration for the 45 load cases.

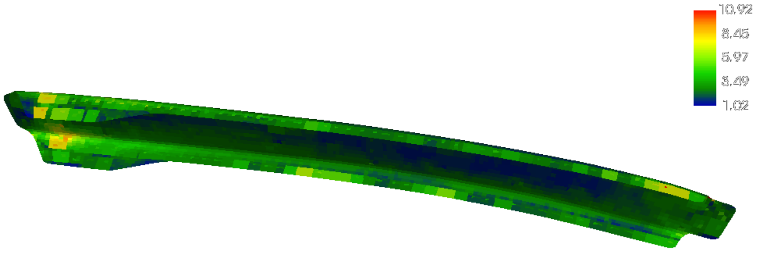


Figure 10. Number of plies after optimization

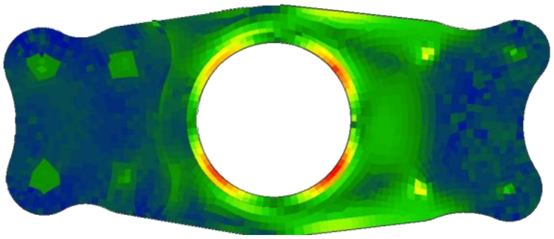


Figure 11. Thickness of metallic component after optimization

The optimizer can hence consider all 45 different scenarios, is satisfying multiple constraints for each material and provides as result a structure which is able to carry the applied load cases and whose parts are not oversized at the same time, meaning that the result is as light as possible. In total 495 evaluations of the full model are necessary to achieve this result which compared to other optimization algorithms is more than competitive.

7. References

1. R. Kriechbaum, J. Schafer and M. Mattheck. CAIO (computer aided internal optimisation): A powerful method to optimise fiber arrangement in composites. In First European conference on smart structures and materials, 281-284, 1992
2. Z.D. Ma, N. Kikuchi, C. Pierre and B. Raju. Multidomain topology optimization for structural and material design. ASME Journal for Applied Mechanics 73(4): 565–573, 2006
3. Dassault Systèmes. Abaqus 6.12 Analysis User's Manual.
4. E. Verginer. New methods for lay-up optimization of composites, Master Thesis, Innsbruck 2014
5. A. Kofler. Local and global smoothing of data for lay-up optimization of composites, Master Thesis, Innsbruck 2014

8. Acknowledgements

This financial support of the Austrian Space Applications Programme of the Austrian Ministry for Transport, Innovation and Technology (BMVIT) is deeply appreciated by the authors.

The authors also thank A. Kofler and E. Verginer, diploma students at the University of Innsbruck, for their contributions to the paper.