NTNU
Norwegian University of Science and Technology

Department of mechanical and industrial engineering

TMM4175 Polymer Composites

AS5 - Portable composite bridge for medium size SUV

Authors
Alberto Da Rold
Vegard G. Jervell
Erwan Luguern
Cyriaque Mas

Supervisor
Nils-Petter Vedvik

May 4, 2021

Contents

1 Introduction

In this report it is described the design of a lightweight portable composite bridge for small to medium size SUV cars. The bridge is made of two beams of 10 meters length, made of the construction and same geometry of the cross section. Each beam is intended to be portable in the sense that two person could carry one of them without problem. It is assumed that the maximum weight that can lift is around 30 kg. So the main objective during the design phase will be to realize a beam that weight approximately 60kg.

However the bridge need to be able to handle the weight of the car which is 2000kg distributed equally between the 2 profiles. The wheelbase of the vehicle is supposed to be the minimum of the range that is proposed since the worst case scenario is the one where all the weight is concentrated in one point.

The global coordinate system for the vessel is indicated in Figure 1.1. Further, throughout the entire analysis it is assumed that laminate layers are perfectly bonded and homogeneous, and free edge effects are neglected.

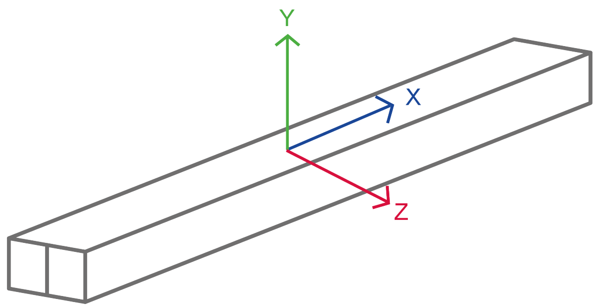


Figure 1.1: Coordinate system of the beam

1.1 Material properties

The beam was designed first using only a carbon/epoxy composite with properties as shown in Table 1.1, thereafter a sandwich structure employing a cross-linked PVC foam with properties as shown in Table 1.1 was used to improve the design.

Table 1.1: Characteristic properties of materials employed in beam design.

Carbon/Epoxy (a)					
Property	Value [MPa]	Property	Value [MPa]	Property	Value [-]
E_1	130000	X_T	1800	ν_{12}	0.28
E_2	10000	Y_T	40	f_{12}	-0.5
G_{12}	4500	X_C	1200	ρ_L	$1.6\,\mathrm{kg\,mm^{-3}}$
S_{23}	40	Y_C	180		
S_{12}	70				
Foam, cross linked PVC					
E [MPa]	100	ν [-]	0.3	ρ	$0.1\,\mathrm{kg\,mm^{-3}}$

2 Theoretical equations

2.1 Loads

The beam needs to support its own weight and the weight of the car. The load from the vehicle can be computed as $P = mg$, with m = 2000 kg and g = 9,81 $m.s^{-2}$, we obtain that P_{tot} = 19620 N. This load will be assumed subdivided equally between the 4 wheels so for each beam we can consider the application of two forces $P = 4905N$ that have a distance from the centre of the beam of $b = 2.5$m as can be represented by the figure 2.1. In the same way, this study will analyse the worse scenario for the beam. Its mean that $a = 3,75$m. The gravity load is distributed along the beam. The constant used in this formula are explained in Figure 2.1.

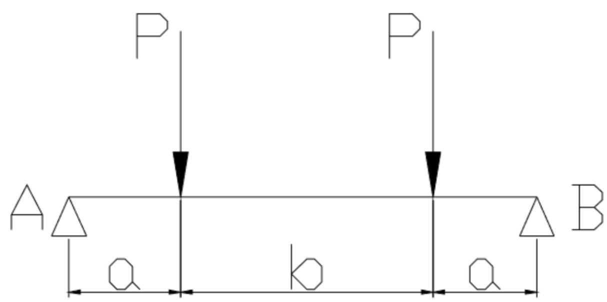


Figure 2.1: Parameters used in computation of beam loading.

The maximum momentum at the center can be computed using this equation given by the literature [1]

$$M_{max} = Pa \tag{2.1}$$

In the previous computation, the weight of the beam is neglected. The following computation proves that its influence can be neglected. Considering that the weight of the beam is 60kg and that it is evenly distributed as a pressure Q, we obtain that the maximum moment in the beam at $x = \frac{L}{2}$ is

$$\begin{aligned} M_{max} &= Pa + Q(\frac{L}{2}x - L) \\ M_{max} &= 15\,480\,\mathrm{N\,m}. \end{aligned} \tag{2.2}$$

Without the weight, the maximum moment is,

$$\begin{aligned} M_{max} &= Pa \\ M_{max} &= 15\,450\,\mathrm{N\,m}. \end{aligned} \tag{2.3}$$

The weight influence is equal to 0,2% and can be thereby neglected, even for weights far exceeding 60 kg.

2.2 2D failure criterion

Here, in this case the Hashin 2D criterion is possible to use, in fact, the force applied on the beam is only following the x and y axis. It is noted that the resulting 2D-criterion may be erroneous at edges, as the plane stress assumption is generally not valid at free edges.

$$f_{FF} = \max\left(\frac{\sigma_1}{X_T}, -\frac{\sigma_1}{X_C}\right) \tag{2.4}$$

for fiber failure criterion.

$$f_{IFF} = \left(\frac{\sigma_2}{Y_T}\right)^2 + \left(\frac{\tau_{12}}{S_{12}}\right)^2, \qquad \sigma_2 > 0, \tag{2.5}$$

for tensile loads, and

$$f_{IFF} = \left(\frac{\sigma_2}{2S_{12}}\right)^2 + \left(\frac{\tau_{12}}{S_{12}}\right)^2 + \left[\left(\frac{Y_C}{2S_{12}}\right)^2 - 1\right]\frac{\sigma_2}{Y_C}, \qquad \sigma_2 < 0 \tag{2.6}$$

for compressive loads. In this analysis, both inter-fiber failure and fiber-failure will be regarded as ultimate failure of the material.

2.3 Stiffness Matrix

For computing the stiffness matrix of the laminate layup, the following computations are used. The compliance matrix is calculable with the relation

$$\mathbf{S} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix}. \tag{2.7}$$

The 2D stiffness matrix is the inverse of the compliance matrix.

$$\mathbf{Q} = \mathbf{S}^{-1} \tag{2.8}$$

Now it is possible to include the orientation of the matrix with the transformation,

$$\mathbf{Q}' = \mathbf{T}_\sigma^{-1}\mathbf{Q}\mathbf{T}_\epsilon. \tag{2.9}$$

With the two transformation matrix expressed as

$$\mathbf{T}_\sigma(\theta) = \begin{bmatrix} c^2 & s^2 & 2cs \\ s^2 & c^2 & -2cs \\ -cs & cs & c^2 - s^2 \end{bmatrix}; \mathbf{T}_\varepsilon(\theta) = \begin{bmatrix} c^2 & s^2 & cs \\ s^2 & c^2 & -cs \\ -2cs & 2cs & c^2 - s^2 \end{bmatrix} \tag{2.10}$$

Where $c = \cos\theta$ and $s = \sin\theta$. The Q matrix components are described as

$$\mathbf{Q}' = \begin{bmatrix} Q_{xx} & Q_{xy} & Q_{xs} \\ Q_{xy} & Q_{yy} & Q_{ys} \\ Q_{xs} & Q_{ys} & Q_{ss} \end{bmatrix}. \tag{2.11}$$

To compute **A** from begin **Q**, laminates loads and constitutive relations are used

$$\mathbf{A} = \sum_k \mathbf{Q}'_k(h_k - h_{k-1}) \tag{2.12}$$

where h_i is the distance from the center of the laminate to the top of layer i.

2.4 Linear Buckling Analysis

Buckling is one the main issue that the analytical approach cannot face but that is possible to verify through the use of FEA. Linear buckling is the most common type of analysis and is easy to execute, but it is limited in the results it can provide.

Linear-buckling analysis calculates buckling load magnitudes that cause buckling and associated buckling modes. Abaqus can provide calculations of a large number of buckling modes and the associated buckling-load factors (BLF). The BLF is expressed by a number which the applied load must be multiplied by to obtain the buckling-load magnitude.

The buckling mode presents the shape the structure assumes when it buckles in a particular mode, but says nothing about the numerical values of the displacements or stresses. The numerical values can be displayed, but are merely relative. This is in close analogy to modal analysis, which calculates the natural frequency and provides qualitative information on the modes of vibration (modal shapes), but not on the actual magnitude of displacements.

Theoretically, it is possible to calculate as many buckling modes as the number of degrees of freedom in the FEA model. In this case we looked only the first positive buckling mode and its associated BLF. This is because higher buckling modes have no chance of taking place — buckling most often causes catastrophic failure or renders the structure unusable.

The nomenclature is "the first positive buckling mode" because buckling modes are reported in the ascending order according to their numerical values. A buckling mode with a negative BLF means the load direction must be reversed (in addition to multiplying by the BLF magnitude) for buckling to happen.

2.5 Safety factor

A safety factor must be chosen for the design of the beam. This factor will be used to intentionally build the beam stronger than required for normal use, thereby giving enough resistance to the beam for emergency situations, unexpected loads, misuse, or degradation. Usually, safety factors are determined by experimentation. This is obviously impossible in the scope of this project. The safety factor has to be chosen with the parameter given.

Here, the problem is a static problem. The main problem here is that we have no information on how the condition will be, wind, bad supports, bad installation. The probability of a misuse are numerous, and a failure is likely to happen if we dimension our system for a vehicle of 2000kg only.

It is commonly accepted that, for a static problem, the safety factor should be included between 2 and 4. In this case, a safety factor of 3 is chosen as a compromise. A too low safety factor will include some risk for the user, and one too big will mean that the system will be over designed. This factor mean that the beam should in principle be able to handle a car of 6000kg. It is estimated that this is enough for all situations that are likely to happen.

3 Analytical analysis

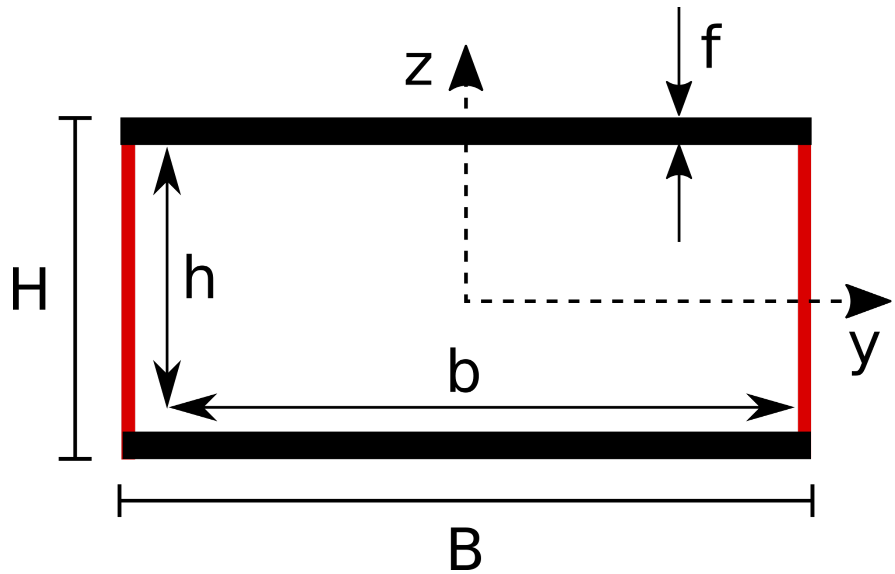


Figure 3.1: Geometry used in analytical analysis. Black and red indicate the top/bottom surface and verticall walls respectively.

An analytical analysis inspired by sandwich theory was implemented in python to acquire a reasonable starting point for the FEA.[2,3] Optimal orientation of the layers in the vertical supports and in the top and bottom faces were determined with respect to minimizing the weight of the bridge while maintaining an exposure factor $f_E < \frac{1}{3}$, where $f_E = \max(f_{FF}, f_{IFF})$ is the Hashin-criterion exposure factor. One layup was chosen for the vertical support walls and another was chosen for the top and bottom surfaces.

The bending stiffness of the beam was computed as

$$D = \int\int z^2 E_{11}(z,y) dzdy \tag{3.1}$$

Where $E_{11}(z, y)$ is the Young's modulus as a function of position. The Young's modulus at any location was approximated as the homogenized Young's modulus of the layup at that position, computed as

$$\bar{E} = \frac{1}{d}\left(A_{xx} - \frac{A_{xy}^2}{A_y y}\right) \tag{3.2}$$

Where d is the thickness of the layup. This gives a beam stiffness of

$$D = \frac{2}{3}\bar{E}_c(B-b)\left(\frac{h}{2}\right)^3 + \frac{2}{3}B\bar{E}_f\left[\left(\frac{H}{2}\right)^3 - \left(\frac{h}{2}\right)^3\right], \tag{3.3}$$

where $\bar{E}_c$ is the homogenized Young's modulus of the vertical walls, $\bar{E}_f$ is the homogenized Young's modulus of the top and bottom surfaces, B and H are the outer width and height of the beam, and b and h are the inner width and height of the beam, as shown in Figure 3.1.

The stress in the top and bottom faces are assumed to be aligned with the x-axis, and can be computed as

$$\sigma_{x,f} = \frac{z\bar{E}_f M_{max}}{D} \tag{3.4}$$

where M_{max} is the internal moment, as computed in section 2.1. The stress in the vertical walls is assumed to consist of a stress component σ_x and a shear component τ_{xz}, these can be computed as

$$\sigma_{x,c} = \frac{z\bar{E}_c M_{max}}{2D}, \qquad \tau_{xz,c} = \frac{Q_x}{2D}\left[\bar{E}_c\left(\left(\frac{h}{2}\right)^2 - z^2\right) + \bar{E}_f f(f+h)\right], \tag{3.5}$$

where $Q_x = \frac{dM_{max}}{dx}$, and f is the thickness of the top and bottom surfaces, as indicated in Figure 3.1. The exposure factor was computed at $z = \{\frac{H}{2}, \frac{h}{2}, 0\}$, as these are the regions with highest compressive strain in the upper face, highest compressive strain in the vertical walls and highest shear in the vertical walls respectively. The exposure factor of the beam was taken to be the maximum of the exposure factor in these regions. Finally, the required vertical wall thickness to achieve an exposure factor $f_E = \frac{1}{3}$ was computed for varying thickness of the upper and lower surfaces using different orientations for the layers in the vertical walls. A flowchart of the algorithm is presented in Figure 3.3, the results of the computation are displayed in Figure 3.2.

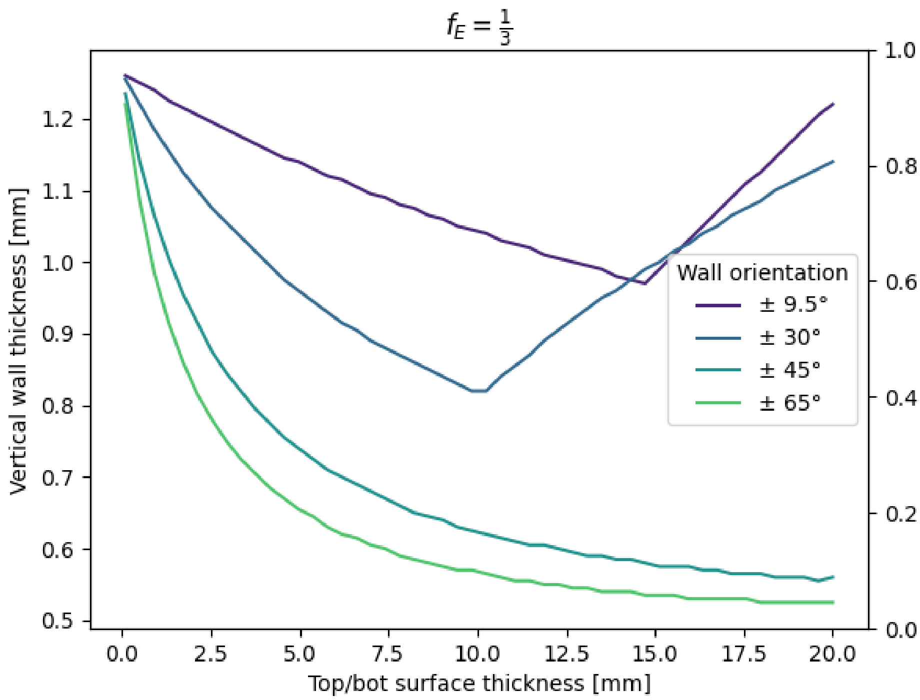


Figure 3.2: Vertical wall thickness required to achieve an exposure factor of $f_E = \frac{1}{3}$ for varying thickness of top and bottom surfaces, using different layup orientations in the vertical walls. The top and bottom surface consist of only 0° oriented layers.

The relationship between the layup orientation in the vertical walls and the maximum deflection of the beam was also computed. Inspired by sandwich theory, the total deflection was taken to be a superposition of bending of the upper and lower surface, and shear deformation of the vertical walls. Treating the load from the vehicle as two point loads P at a distance a from each end of the beam allows the total deflection of the beam to be calculated as a superposition of the deflection resulting from each of these loads. The total,

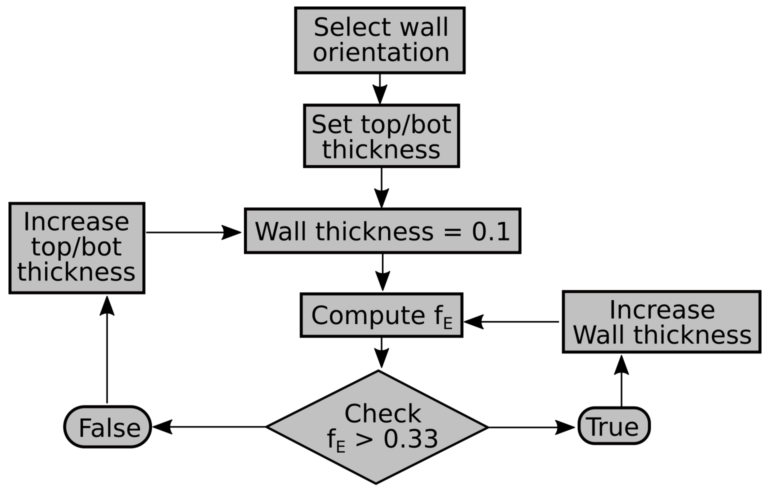


Figure 3.3: Algorithm for computing wall thickness required to achieve an exposure factor $f_E = \leq \frac{1}{3}$, for varying thickness of the top and bottom surface. The top and bottom surfaces were set to consist of only 0° oriented layers.

maximum deflection will then be given as

$$\delta_{max} = 2\left(\frac{Pa}{48D_b}(3L^2 - 4a^2) + \frac{Pa}{2S}\right), \tag{3.6}$$

Where S is the shear stiffness of the vertical walls, and D_b is the bending stiffness of the top and bottom surface. For a sandwich structure, S is computed as $S = bhG_c$ where G_c is the the shear modulus of the core material. For the box beam considered here, G_c is taken to be the effective shear modulus of the vertical walls, computed as

$$G_c = \frac{A_{ss}}{B} \tag{3.7}$$

where A_{ss} is computed for the two vertical walls, separated by a "layer" of air with thickness b, with all elements in the 2D-stiffness matrix equal to zero. The maximum deflection as a function of layup orientation in the vertical walls and as a function of layup orientation in the top and bottom surfaces is displayed in Figure 3.4.

Due to the possibility of buckling occurring, a combination of layup orientations that minimizes the deformation was chosen as the starting point for the FEA analysis. As seen from Figure 3.4, this is achieved for layers oriented ±9.5° in the vertical walls, and 0° in the top and bottom surfaces.

The initial thickness of the vertical walls and the top and bottom surfaces was chosen by inspecting Figure 3.2. According to the figure, a top and bottom surface thickness of 5 mm and vertical wall thicknesses of 1.1 mm will give a maximum exposure factor of $\frac{1}{3}$. Figure 3.2 also shows that this is not, per the analytical analysis, the combination that will give the lowest weight while maintaining a safety factor of 3. However, solutions with side wall thickness below 1 mm are expected to produce buckling.

3.1 Design of the profile's cross section

It is noted that the analytical analysis that was implemented does not take into consideration the horizontal position of the vertical walls. In practice this means that the analytical approach does not differentiate between the geometry shown in Figure 3.5, with five vertical supports, and the geometry shown in Figure 3.1 with only two, given that the total thickness of the vertical supports is equal. To prevent the top surface

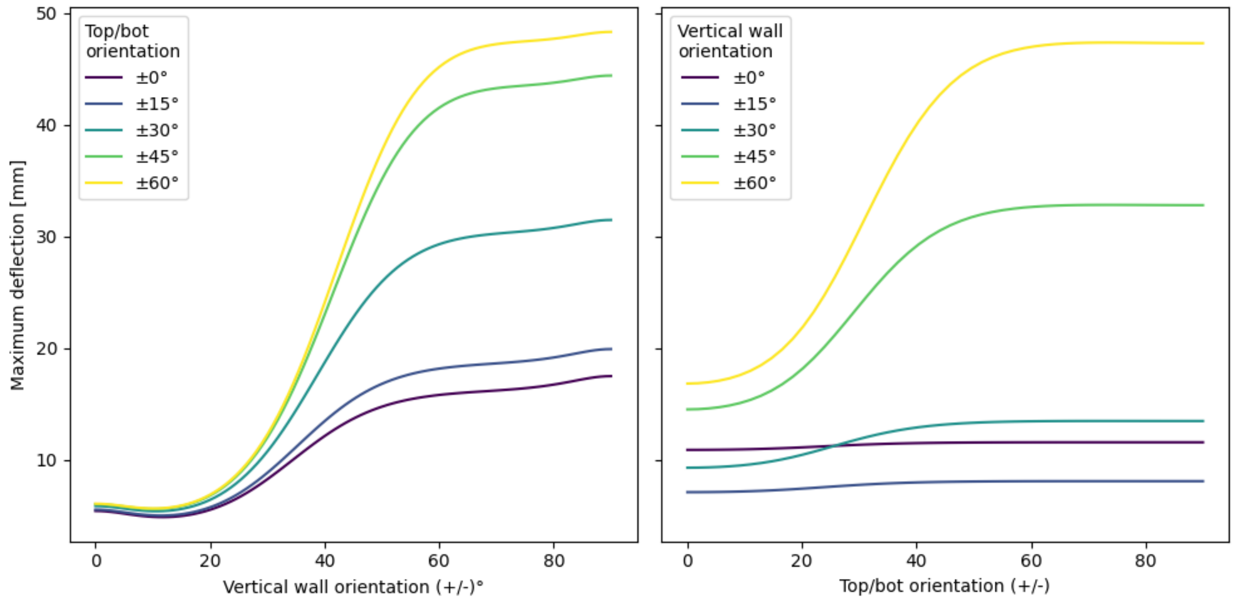


Figure 3.4: Maximum beam deflection as a function of layup orientation in the vertical walls (left) and in the top and bottom surfaces (right). Values as computed from equation (3.6).

from breaking due to bending in the transverse direction of the beam, a geometry with five vertical supports, as shown in Figure 3.5, with a total thickness 2.2 mm was used as the initial configuration.

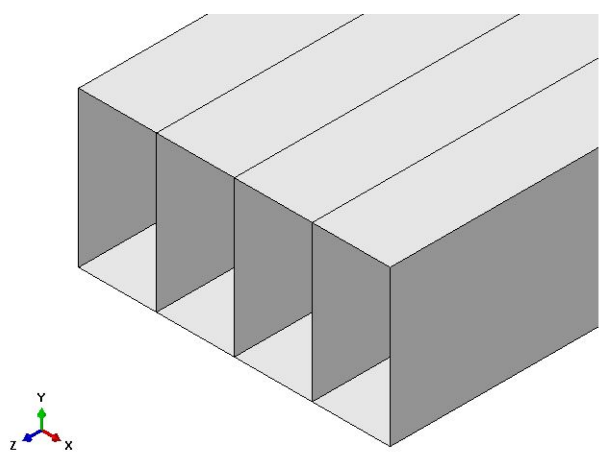


Figure 3.5: Internal topology of the beam. Outer dimensions are 500x250mm

4 Finite Element Analysis

Using the results from the analytical analysis as an initial configuration, a finite element analysis was conducted with the goal of minimizing the weight of the beam while maintaining a Hashin-criterion exposure factor $f_E < \frac{1}{3}$. The major issue that can be handled by FEA which was not included in the analytical analysis is that of buckling in the vertical supports and in the top surface. Using FEA, the viability of a sandwich structure was also investigated. The final results of the finite element analysis beams B and D presented in Table 4.1. The bridge utilizing sandwitch structures has a far lower mass, but is extremely fragile if handled incorrectly, as discussed in detail in sections 4.3 and 5.

Table 4.1: Notable beam structures investigated by finite element analysis. The letter "f" indicates a foam layer with properties as described in Table 1.1. Beam A is the initial configuration chosen based on the analytical analysis, beam B is the result of optimization without a sandwich solution. Beam C is the initial configuration used for optimization with a sandwich solution, beam D is the final result of optimization with a sandwich solutions.

Beam		Top Surface	Bottom Surface	Vertical Supports	Eigenvalue, 1st buckling mode	$f_E^{\max}$	Mass [kg]
A	Orientation	$[0]^{10}$	$[0]^{10}$	$[9.5, -9.5]_s$	0.1	0.0003	168
	Thickness	$[1]^{10}$	$[1]^{10}$	$[0.22, 0.22]_s$			
B	Orientation	$[0, 90, 0, 90]_s$	$[0]^6$	$[9.5, -9.5]_s$	1.36	0.002	91.8
	Thickness	$[0.5, 0.2, 1.5, 0.1]_s$	$[0.5]^6$	$[0.4, 0.4]_s$			
C	Orientation	$[0, 90, 0, 90, f]_s$	$[0]^6$	$[9.5, -9.5, f]_s$	7.1	0.002	57.6
	Thickness	$[0.5, 0.2, 1.5, 0.1, 5]_s$	$[0.5]^6$	$[0.4, 0.4, 5]_s$			
D	Orientation	$[0, 90, f]_s$	[0]	$[9.5, -9.5, f]_s$	1.70	0.03	57.6
	Thickness	$[0.4, 0.1, 5]_s$	[0.6]	$[0.1, 0.1, 5]_s$			

4.1 Model characteristics

ABAQUS Version 6.21 was used in order to compute the FEA analysis. The beam is simply supported at the extremities and distributed load due to gravity and two concentrated loads as consequence of the load of the car were applied. The total mass that has to handle one single beam is 1000kg (500kg per wheel), in order to simulate it, 6 masses of 167kg were added in 6 different point (3 for each wheel) as can be seen from figure 4.1

Different layup are applied for the different part of the beam as can be seen from 4.2. The supports will has the same layup but the two lateral walls has the offset set to *top* while the central supports has the offset set to *middle* For the element meshing it was used a mesh size of 50 and a *Quad* shape combined with a *Structured* technique.

4.2 Plain laminate structure

As shown in Figure 4.3, the vertical walls in the initial configuration buckle at very low loads, with a buckling eigenvalue of 0,78166. Additionally, Figure 4.4 shows that the analytical analysis over-predicted the Hashin-criterion exposure factor.

Based on these results the thickness of the vertical walls was increased, while the thickness of the top and bottom surfaces was decreased. It is also noted that the bottom surface is only subjected to a tensile load therefore it will not buckle. Furthermore, it does not need to support the concentrated load of the wheels of the vehicle. Due to these considerations the bottom layer was chosen to be notably thinner than the top layer. The thickness of the vertical supports was increased until the first positive buckling mode was buckling

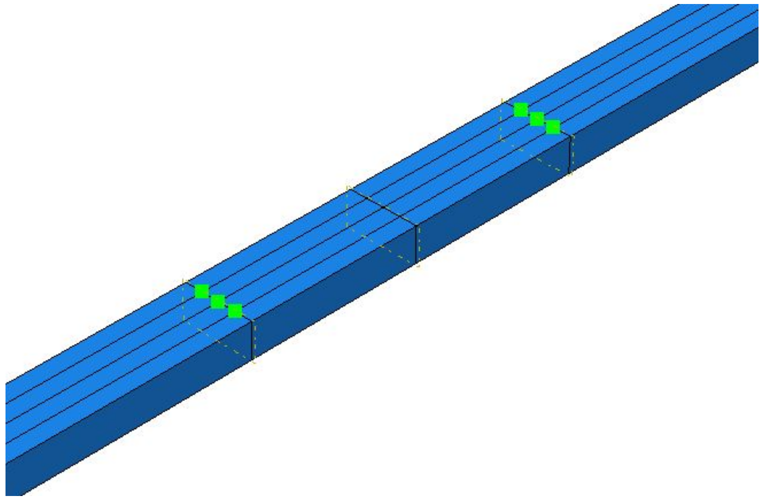


Figure 4.1: Mass distribution on the top of the beam in correspondence with the wheels.

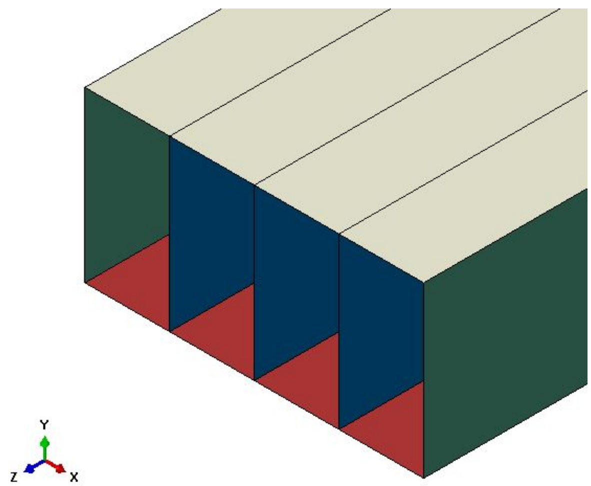


Figure 4.2: Beam section with the different layup applied.

in the top surface, with an eigenvalue of 1.36, as shown in Figure 4.5. Further, a small number of 90° layers were introduced in the top surface to prevent deformation between the vertical supports. As shown in Figure 4.6 the Hashin-criterion exposure factor was far below one.

Varying the orientation of the layers in the vertical walls to increase their resistance to buckling was attempted, but the reduction in stiffness lead to increased deformation and buckling in the top surface. Comparing the analytical value of the maximum deflection in left graph of figure 3.4 with the value that results from the FEA (4.7), considering an angle of the fiber of the vertical walls of ± 45, we are able to verify that the results obtained with FEA regarding deformation are consistent with the analytical approach since the maximum value is approximately $31mm$ in both cases.

4.3 Sandwich structure

To enable further reduction of the thickness of the laminates, the beam was modeled using sandwich structures. The sandwich structure prevents buckling, but does not increase the tensile strength of the walls. In

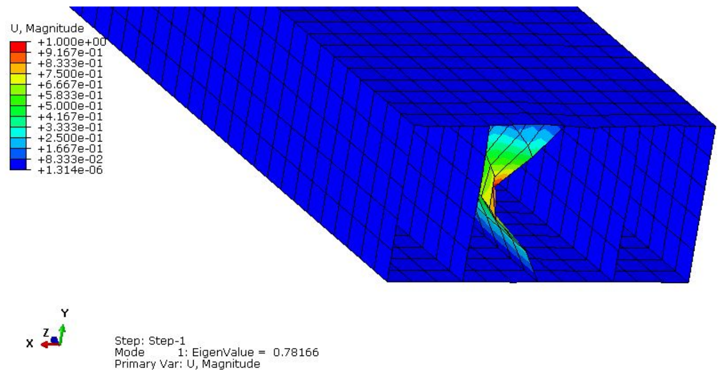
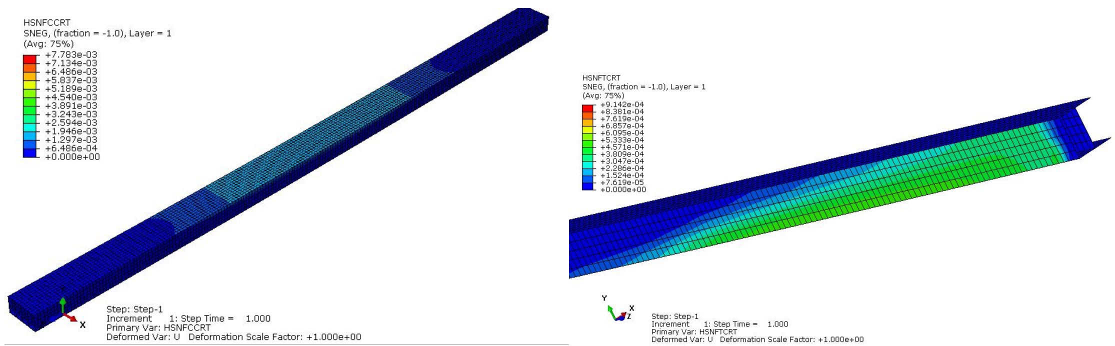


Figure 4.3: Buckling occurs in the vertical walls if they are thinner than 1.6 mm, with top and bottom surfaces of thickness 10 mm.



(a) Compressive (b) Tensile

Figure 4.4: Hashin-criterion exposure factor for beam A, shown in Table 4.1. Exposure factors are in the order of 10^{-3}.

fact, as can be seen from equation (3.3), the bending stiffness of the beam will be decreased if sandwich structures are used for the top and bottom surfaces while the total thickness of laminate is kept constant. Therefore, a sandwich structure was only used in the vertical supports and the top surface, as these are the only regions exposed to buckling.

Initially, a foam layer of 10 mm was introduced in the top surface and vertical walls while keeping the thickness of the Carbon-Epoxy constant. This lead to a greatly reduced eigenvalue of the first buckling mode as shown in Figure 4.8. The increased resistance to buckling allowed for reduction in the thickness of the laminates.

The thickness of the laminate faces in the sandwich structured walls was decreased until the eigenvalue of the first mode of buckling was 1.34. Further reduction of the thickness of the laminate layers in the vertical supports was not deemed viable due to production limitations. Reduction of the foam thickness in the top surface lead to buckling modes with eigenvalues below one. The final configuration using sandwich panels was thereby determined to be the structure shown in Figure 4.10.

It is noted that the extremely thin bottom layer is incapable of carrying any significant load. A buckling analysis shows that there is a buckling mode with eigenvalue -0.02, as shown in Figure 4.9. This indicates that the beam is likely incapable of supporting its own weight if carried or placed upside down.

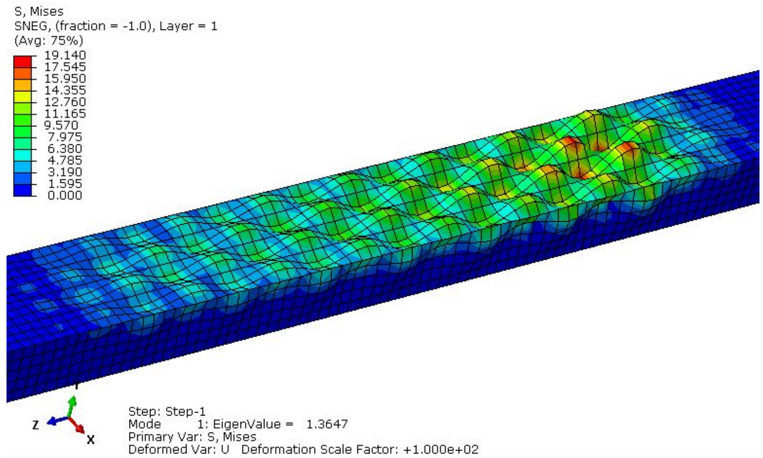
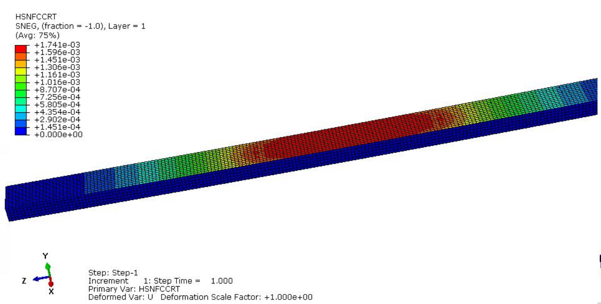
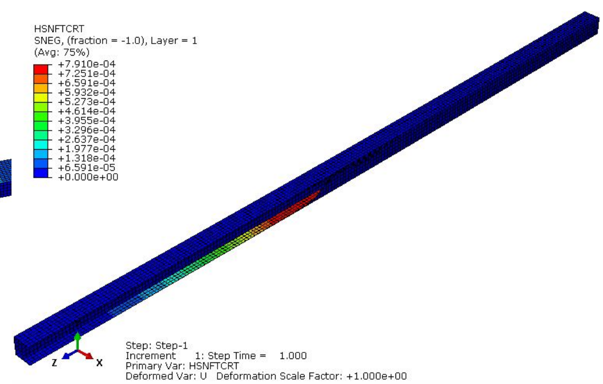


Figure 4.5: First buckling mode for a beam with 1.6 mm thick vertical supports oriented ±9.5°, 4.6 mm thick top surface and 3 mm thick bottom surface, with orientations as shown for beam B in Table 4.1.



(a) Compressive



(b) Tensile

Figure 4.6: Hashin-criterion exposure factor for a beam with 1.6 mm thick vertical supports oriented ±9.5°, 4.6 mm thick top surface and 3 mm thick bottom surface, with orientations as shown for beam B in Table 4.1. Maximum exposure factor is 0.002.

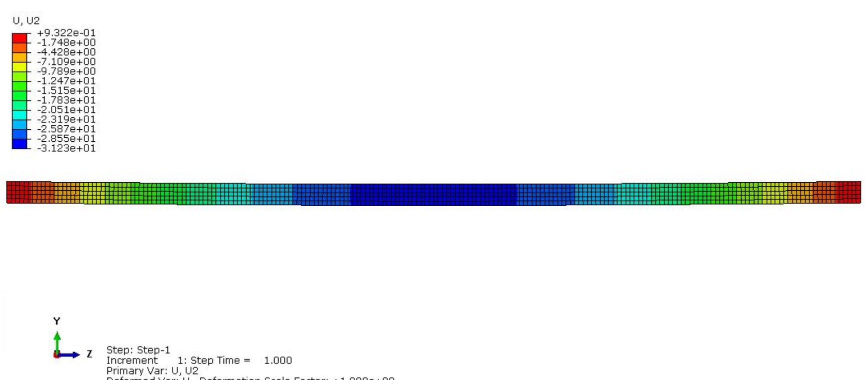


Figure 4.7: Beam deflection for a angle of ±45° of the vertical wall, with all other properties equal to those of beam B, presented in Table 4.1. Maximum deflection is ≈30 mm.

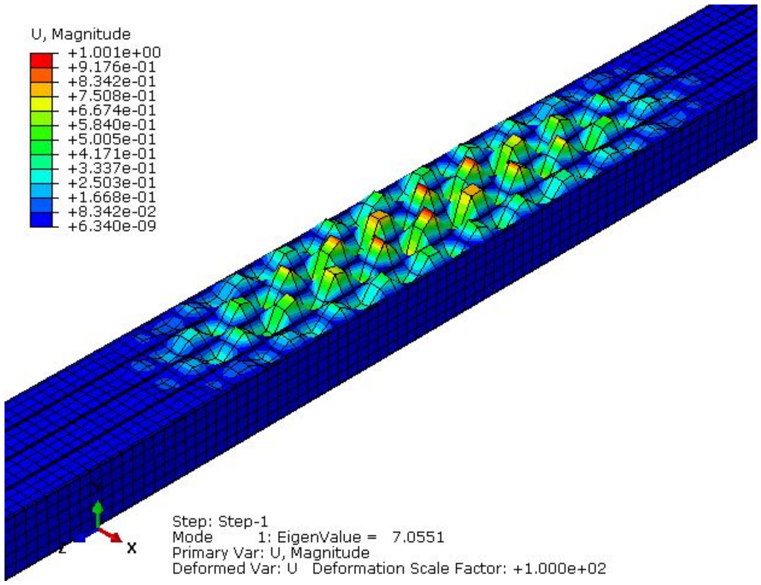


Figure 4.8: First positive buckling mode for beam C, presented in Table 4.1.

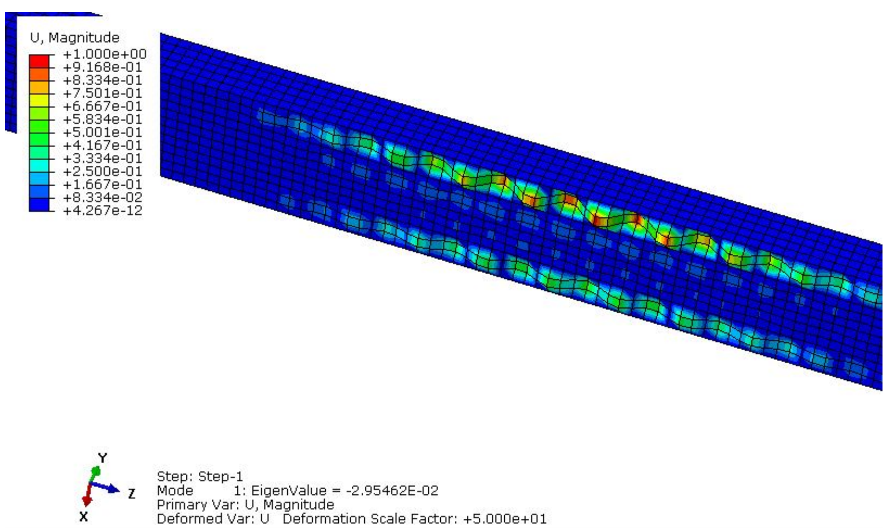


Figure 4.9: First mode with BLF negative on the bottom of the beam, for beam D, presented in Table 4.1

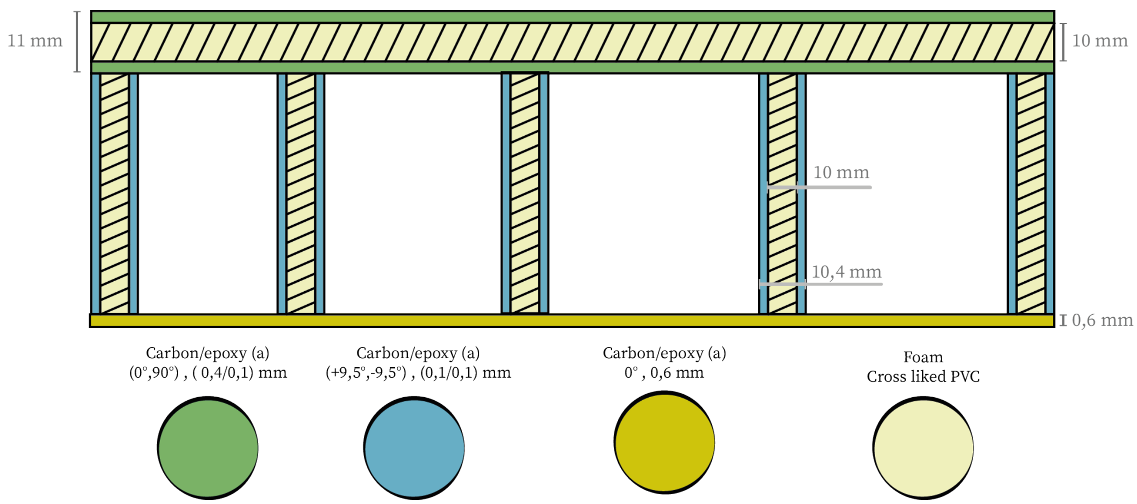


Figure 4.10: Cross-section of beam D, presented in Table 4.1.

5 Discussion

Assumption were made in this design process, and they can probably modify the real behavior of the proposed beam. First, in the FEA, the tire pressure have been transformed into 3 force applied on 3 point for each wheel. This allow us to reduce the complexity of the FEA model and to have faster results. However, a tire is a flexible surface, and a better approximation of its force would be a rectangular region where a pressure is applied. To create such a model, more data regarding the properties of the tire would be required. A better tire simulation can probably change some results, the pressure of the vehicle can be better divided between the support walls and in this case reduce the buckling and the deformation observed during the FEA. In this regard, treating the load from the tires as a series of point loads can be regarded as a safety measure.

Now, looking at the results through the reality perspective, some interesting points can be raised. All the laminates are very thin. The FEA shows that they will handle the loads and forces applied on them, but considerations regarding different damage that can happen during production or transport have not been regarded. Such a bridge will probably be exposed to extreme condition, add a protective shell for the composite or a thicker layup should be considered. This shield layer can increase the mechanical properties of the beam and protect from rock, humidity, heat, or an uneven ground when setting up the bridge.

It is also important to note that the bridge is not symmetrical in the vertical direction. This has facilitated a greatly reduced weight, as the thickness of the bottom surface could be drastically reduced. However, this also implies that the capacity of the bridge will be massively reduced if it is placed the wrong way. If the beam is to be produced, the surfaces must be clearly marked to indicate which side is up, as it will likely fail under its own weight if transported or carried upside down. To mitigate this problem, a custom-made holder that properly secures the beam to the vehicle, and provides support during transportation could be created. Local reinforcement of the ends of the beam may also be a viable way to reduce the danger of breakage during transportation and mounting.

It is possible to test the buckling in a an analytical way, nonetheless this analysis include to solve a second-order, linear ordinary differential equation [4]. The expense of time to solve such an equation has been regarded to be is too long compared to the increased precision. Thus, the FEA has been judged to be enough.

5.1 Manufacturing

The manufacturing processes for sandwich materials can be the same as those used for other composite materials or specific processes. For manufacturing the solution chosen. The more efficiency way seems to create each sandwich and then assemble them together using plastic welding or addition of joints in each junction.

Each sandwich part can be created using wet or dry moulding techniques. For wet moulding could use contact molding, vacuum molding, compression molding could be used. For a dry moulding, it is possible to use preimpregnated laminates. Glue technique could be used as well, however to determine which assembly technique is the best for our composite, experimentation would be required. This assembly can modify the mechanical properties, some further research would have to be done if the beam were to be produced.

6 Conclusion

This report show all the thinking and the step through the creation of this car bridge beam. An analytical analysis has been created, using this algorithm one solution have been chosen. Then this solution has been processed through Abaqus software to confirm that the analytical result were valid and to further optimize the solution while taking into account buckling.

On a deformation and classic material failure, the analytical solution fulfilled all the criterion. However, the buckling analysis showed a huge risk of breakage. Even if, like described in part 2.4, we have no actual information on the magnitude of displacements. With some experiments, it could have been possible to

Table 6.1: Optimal beam structures determined by finite element analysis. The letter "f" indicates a foam layer.

Beam		Top Surface	Bottom Surface	Vertical Supports	Eigenvalue, 1st buckling mode	$f_E^{\max}$	Mass [kg]
B	Orientation	$[0, 90, 0, 90]_s$	$[0]^6$	$[9.5, -9.5]_s$	1.36	0.002	91.8
	Thickness	$[0.5, 0.2, 1.5, 0.1]_s$	$[0.5]^6$	$[0.4, 0.4]_s$			
D	Orientation	$[0, 90, f]_s$	[0]	$[9.5, -9.5, f]_s$	1.70	0.03	57.6
	Thickness	$[0.4, 0.1, 5]_s$	[0.6]	$[0.1, 0.1, 5]_s$			

determinate these magnitudes and to decide if this solution was feasible or not. Further optimization by FEA resulted in beam B, presented in Table 6.1 with a greatly reduced weight.

For the second solution, a sandwich structure has been developed. The same geometry is kept, but with a different layups. This structure is based on two skin of carbon/epoxy (a) composite and a core of PVC foam. The structure resulting from the optimization without a sandwich solution was used as an initial configuration, before further optimization with a sandwich solution. The new solution this time validate all the constraint of the problem and is even better than the last one with a lighter design.

The two solutions presented in Table 6.1 are therefore optimal for different conditions. Beam B is more robust with regard to handling and transportation, but with a weight below 100 kg it can still be regarded as portable. Beam D is optimized for low weight, and can easily be transported by a single person if one end is mounted on wheels, however it is extremely fragile if handled incorrectly, and may be poorly suited to be mounted on uneven ground. This may be solved by good design of the end ramps or local reinforcement of the ends.

This solution has now to be assembly with the ramp to the two extremity and tested in real conditions.

References

[1] N. franceserv, "Formulaire poutres simples." http://notech.franceserv.com/formulaire-des-poutres.pdf.

[2] Divinycell, Klegecell, and Probalsa, "DIAB sandwich handbook." http://www.stmboats.com/articles/sandwich_hb.pdf.

[3] Wikipedia, "Sandwich theory." https://en.wikipedia.org/wiki/Sandwich_theory.

[4] Boston university, "Mechanics of materials: Beam buckling." http://www.bu.edu/moss/mechanics-of-materials-beam-buckling/.

A Python code

A.1 main.py

```
'''
Purpose: Find optimal orientation of laminates to minimize deflection of box beam
Requires: Numpy, matplotlib
'''

import laminate as lam
import constants as c
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.cm import get_cmap

from laminate import get_sandwitch_D, get_eff_G, get_E_hom


def gen_air_width(side_layup):
    h = c.B - 2 * lam.laminateThickness(side_layup)
    return [{'mat' : c.no_mat, 'ori': 0, 'thi': h}]

def gen_air_height(tb_layup):
    h = c.H - 2 * lam.laminateThickness(tb_layup)
    return [{'mat' : c.no_mat, 'ori': 0, 'thi': h}]

def get_deformation(tb_layup, side_layup):
    Db = get_sandwitch_D(tb_layup, side_layup)
    S = get_eff_G(2 * side_layup) * lam.laminateThickness(2 * side_layup) * (c.H - lam.laminateThickness(tb_layup))

    # From http://www.stmboats.com/articles/sandwich_hb.pdf
    d1 = c.F * c.a * (3 * c.L**2 - 4 * c.a**2) / (48 * Db)
    d2 = c.F * c.a / (2 * S)

    return 2 * (d1 + d2)

def hashin(stress):
    #Compute hashin criterion exposure factor, return max(fE_FF, fE_IFF)
    s1, s2, s12 = stress
    XT, YT, XC, YC, SL, ST = c.m['XT'], c.m['YT'], c.m['XC'], c.m['YC'], c.m['S12'], c.m['S23']

    fE_FF = max((s1/XT, -s1/XC))

    if s2 >= 0:
        fE_IFF = (s2/YT)**2 + (s12/ST)**2
    else: # s2 <0:
        fE_IFF = ((s2/(2 * ST))**2 + ( (YC/(2 * ST))**2 - 1) * (s2/YC) + (s12/ST)**2)

    return max((fE_FF, fE_IFF))

def plot_ratio_tb_deflection ():
    ratios = np.linspace(0, 1, 50)
    side_layup = [{'mat': c.m, 'ori': 45, 'thi': 0.25},
                  {'mat': c.m, 'ori': -45, 'thi': 0.5},
                  {'mat': c.m, 'ori': 45, 'thi': 0.25}]
    deflections = np.zeros(len(ratios))
    for i, r in enumerate(ratios):
        tb_layup = [{'mat': c.m, 'ori': 0, 'thi': r/2},
                    {'mat': c.m, 'ori': 90, 'thi': (1 - r)},
                    {'mat': c.m, 'ori': 0, 'thi': r/2}]
        deflections[i] = get_deformation(tb_layup, side_layup)

    plt.plot(ratios, deflections)
    plt.xlabel('Top/bot ratios (0/90)')
    plt.ylabel(r'$\delta$')
```

```
    plt.show()

def plot_ori_side_deflection(tb_ori=10):
    tb_layup = [{'mat': c.m, 'ori': tb_ori, 'thi': 1},
                {'mat': c.m, 'ori': -tb_ori, 'thi': 2},
                {'mat': c.m, 'ori': tb_ori, 'thi': 1}]

    t_list = np.linspace(0, 90, 100)
    deflections = np.zeros(len(t_list))
    for i, t in enumerate(t_list):
        side_layup = [{'mat': c.m, 'ori': t, 'thi': 1},
                      {'mat': c.m, 'ori': -t, 'thi': 2},
                      {'mat': c.m, 'ori': t, 'thi': 1}]
        deflections[i] = get_deformation(tb_layup, side_layup)

    plt.plot(t_list, deflections, label='  '+str(tb_ori)+'\u00B0', color=cmap(tb_ori/60))
    plt.xlabel('Vertical wall orientation (+/-)\u00B0')
    plt.ylabel(r'Maximum deflection [mm]')

def plot_ori_tb_deflection(side_ori):
    side_layup = [{'mat': c.m, 'ori': side_ori, 'thi': 0.25},
                  {'mat': c.m, 'ori': -side_ori, 'thi': 0.5},
                  {'mat': c.m, 'ori': side_ori, 'thi': 0.25}]

    t_list = np.linspace(0, 90, 50)
    h = 1
    deflections = np.zeros(len(t_list))
    for i, t in enumerate(t_list):
        tb_layup = [{'mat': c.m, 'ori': t, 'thi': 1},
                    {'mat': c.m, 'ori': -t, 'thi': 2},
                    {'mat': c.m, 'ori': t, 'thi': 1}]
        deflections[i] = get_deformation(tb_layup, side_layup)

    plt.plot(t_list, deflections, label='  '+str(side_ori)+'\u00B0', color=cmap(side_ori/60))

def plot_ratio_side_deflection():
    ratios = np.linspace(0, 1, 50)
    tb_layup = [{'mat': c.m, 'ori': 0, 'thi': 0.25},
                {'mat': c.m, 'ori': 90, 'thi': 0.5},
                {'mat': c.m, 'ori': 0, 'thi': 0.25}]
    deflections = np.zeros(len(ratios))
    for i, r in enumerate(ratios):
        side_layup = [{'mat': c.m, 'ori': 9.5, 'thi': r / 4},
                      {'mat': c.m, 'ori': -9.5, 'thi': r / 4},
                      {'mat': c.m, 'ori': -45, 'thi': (1 - r)/6},
                      {'mat': c.m, 'ori': 45, 'thi': (1 - r) * 2 / 3},
                      {'mat': c.m, 'ori': -45, 'thi': (1 - r)/6},
                      {'mat': c.m, 'ori': -9.5, 'thi': r / 4},
                      {'mat': c.m, 'ori': 9.5, 'thi': r / 4}]
        deflections[i] = get_deformation(tb_layup, side_layup)

    plt.plot(ratios, deflections)
    plt.xlabel('Side ratios ')
    plt.ylabel(r'$\delta$')
    plt.show()

def get_max_exposure(h_side, h_tb, side_ori=9.5):
    side_layup = [{'mat': c.m, 'ori': side_ori, 'thi': h_side/4},
                  {'mat': c.m, 'ori': -side_ori, 'thi': h_side/2},
                  {'mat': c.m, 'ori': side_ori, 'thi': h_side/4}]

    tb_layup = [{'mat': c.m, 'ori': 0, 'thi': 0.455 * h_tb},
                {'mat': c.m, 'ori': 90, 'thi': 0.05 * h_tb},
                {'mat': c.m, 'ori': 0, 'thi': 0.455 * h_tb}]

```

```
    f = lam.laminateThickness(tb_layup)
    w = lam.laminateThickness(side_layup)
    h = c.H - 2 * f
    b = c.B - 2 * w
    D = get_sandwitch_D(tb_layup, side_layup)

    Ef = get_E_hom(tb_layup)
    Ec = get_E_hom(side_layup)

    stress_f_max = - c.F * c.a * c.H * Ef / (2 * D)
    max_stress_f = [stress_f_max, 0, 0]

    stress_c_interface = - c.F * (c.H - h) * c.a * Ec / (2 * D) * (c.B / (c.B - b))
    shear_c_interface = - ( c.F / (2 * D) ) * Ef * f * (f + h) * (c.B / (c.B - b))
    max_stress_c = [ stress_c_interface , 0, shear_c_interface ]

    shear_center = - ( c.F / (2 * D) ) * (Ec * (h / 2)**2 + Ef * f * (f + h)) * (c.B / (c.B - b))
    max_shear = [0, 0, shear_center]

    max_face_fE = 0
    for layer in tb_layup:
        t = layer['ori']
        layer_stress = np.dot(lam.T2Ds(t), max_stress_f)
        this_face_fE = hashin(layer_stress)
        if this_face_fE > max_face_fE:
            max_face_fE = this_face_fE

    max_interface_fE = 0
    max_center_fE = 0
    for layer in side_layup:
        t = layer['ori']
        interface_stress = np.dot(lam.T2Ds(t), max_stress_c)
        center_stress = np.dot(lam.T2Ds(t), max_shear)

        center_fE = hashin(center_stress)
        if center_fE > max_center_fE:
            max_center_fE = center_fE

        interface_fE = hashin( interface_stress )
        if interface_fE > max_interface_fE:
            max_interface_fE = interface_fE

    return max((max_face_fE, max_center_fE, max_interface_fE))

def plot_envelope( side_ori ):
    h_tb_list = np.linspace(5, 10, 50)
    h_side_list = np.zeros(len( h_tb_list ))

    for i, h_tb in enumerate(h_tb_list):
        print(h_tb)
        h_side = 0.5
        expo = get_max_exposure(h_side, h_tb, side_ori=side_ori)
        while expo > 1/c.safety_factor:
            if h_side > 250:
                break
            h_side += 0.005
            expo = get_max_exposure(h_side, h_tb, side_ori=side_ori)

        h_side_list [i] = h_side

    area = 2 * h_tb_list * c.B + 2 * h_side_list * (c.H - 2 * h_tb_list)
    mass = area * c.L * c.m['rho']/1000

    ax.plot( h_tb_list , h_side_list , label='    '+str( side_ori )+'\u00B0', color=cmap(side_ori/65))
    twn.plot( h_tb_list , mass, color=cmap(side_ori/65))
```

```python
    #print('  ', side_ori )

cmap = get_cmap('viridis')
fig, axs = plt.subplots(2,1, sharex='all')
ax, twn = axs

plot_envelope(9.5)
plot_envelope(30)
plot_envelope(45)
plot_envelope(65)

ax.set_ylabel('Vertical wall thickness [mm]')
twn.set_xlabel('Top/bot surface thickness [mm]')
twn.set_ylabel(r'total mass [kg]')
plt.suptitle(r'$f_E = \frac{1}{3}$')
ax.legend(title='Wall orientation')
plt.tight_layout()
plt.show()

fig, axs = plt.subplots(1,2, sharey='all', figsize=(10,5))
plt.sca(axs[0])

plot_ori_side_deflection(0)
plot_ori_side_deflection(15)
plot_ori_side_deflection(30)
plot_ori_side_deflection(45)
plot_ori_side_deflection(60)
plt.legend(title='Top/bot\norientation')

plt.sca(axs[1])
plot_ori_tb_deflection(0)
plot_ori_tb_deflection(15)
plot_ori_tb_deflection(30)
plot_ori_tb_deflection(45)
plot_ori_tb_deflection(60)
plt.legend(title='Vertical wall\norientation')
plt.xlabel('Top/bot orientation (+/-)')
plt.tight_layout()
plt.show()
```

A.2 laminate.py

```python
'''
Purpose: Computing laminate properties, and transformation matrices
Author: Adapted by Vegard G. Jervell from code by Nils Petter Vedvik
URL: https://folk.ntnu.no/nilspv/TMM4175
Requires: NumPy
'''

import numpy as np

import constants as c


def S2D(m):
    return np.array([[        1/m['E1'], -m['v12']/m['E1'],          0],
                     [-m['v12']/m['E1'],         1/m['E2'],          0],
                     [                0,                 0, 1/m['G12']]], float)
def Q2D(m):
    S=S2D(m)
    return np.linalg.inv(S)

def T2Ds(a):
    c,s = np.cos(np.radians(a)), np.sin(np.radians(a))
    return np.array([[ c*c ,  s*s ,   2*c*s],
```

```
                      [ s*s ,  c*c ,  -2*c*s],
                      [-c*s,   c*s , c*c-s*s]], float )

def T2De(a):
    c,s = np.cos(np.radians(a)), np.sin(np.radians(a))
    return np.array ([[    c*c,    s*s,       c*s ],
                      [    s*s,    c*c,      -c*s ],
                      [-2*c*s, 2*c*s, c*c-s*s ]], float )

def laminateThickness(layup):
    return sum([layer['thi'] for layer in layup])

def Q2Dtransform(Q,a):
    return np.dot(np.linalg.inv( T2Ds(a) ) , np.dot(Q, T2De(a)) )

def computeA(layup):
    A=np.zeros((3,3),float)
    hbot = -laminateThickness(layup)/2       # bottom of first layer
    for layer in layup:
        m = layer['mat']
        Q = Q2D(m)
        a = layer['ori']
        Qt = Q2Dtransform(Q, a)
        htop = hbot + layer['thi']    # top of current layer
        A += Qt*(htop-hbot)
        hbot = htop                   # for the next layer
    return A


def get_sandwitch_D(tb_layup, side_layup):
    f = laminateThickness(tb_layup)
    w = laminateThickness(side_layup)
    h = (c.H - 2 * f)
    b = (c.H - 2 * w)

    Ef = get_E_hom(tb_layup)
    Ec = get_E_hom(side_layup)

    D1 = Ef * c.B * (2 / 3) * ( (c.H / 2)**3 - (h / 2)**3)
    D2 = Ec * (2 / 3) * (c.B - b) * (h / 2)**3
    return D1 + D2


def get_eff_G(layup):
    A = computeA(layup)
    return A[2,2]/laminateThickness(layup)


def get_E_hom(layup):
    A = computeA(layup)
    h = laminateThickness(layup)
    return (1/h) * (A[0,0] - ( (A[0,1]**2 / A[1, 1])))


def get_sandwitch_Ixx(tb_layup):
    f = laminateThickness(tb_layup)
    return (c.B * f**2 / 12) + 2 * c.B * f * (c.H - f/2)**2
```

A.3 constants.py

```
'''
Case spesific parameters and material properties
'''

M = 2 #vehicle mass [Mg]
```

```
L = 10e3 # beam length [mm]
w1 = 2.5e3 #wheelbase length [mm]
w2 = 2.5e3 #wheelbase B [mm]
g = 9810 #mm/s^2

H = 250 #mm
B = 500 #mm

F = M * g / 4
a = (L - w1)/2

 safety_factor  = 3

no_mat = {"name": "Air", "units": "MPa-mm-Mg", "type": "None", "fiber": "None",
        "Vf": 1e-10, "rho": 1e-10,
        "description": "Air",
        "E1": 1e-10, "E2": 1e-10, "v12": 1e-10, "v23": 1e-10, "G12": 1e-10,
        "a1": 1e-10, "a2": 1e-10,
        "XT": 1e10, "YT": 1e10, "XC": 1e10, "YC": 1e10,
        "S12": 1e-10, "S23": 1e-10, "f12": 1e-10}

m = {"name": "Carbon/Epoxy(a)", "units": "MPa-mm-Mg", "type": "UD", "fiber": "Carbon",
        "Vf": 0.50, "rho": 1.6e-3,
        "description": "UDFC Carbon/Epoxy",
        "E1": 130000, "E2": 10000, "v12": 0.28, "v23": 0.5, "G12": 4500,
        "a1": 8e-06, "a2": 25e-06,
        "XT": 1800, "YT": 40, "XC": 1200, "YC": 180,
        "S12": 70, "S23": 40}
```