DESIGN OF A COMPOSITE-SANDWICH SKATEBOARD OPTIMIZER

Dino Spiller 1084324

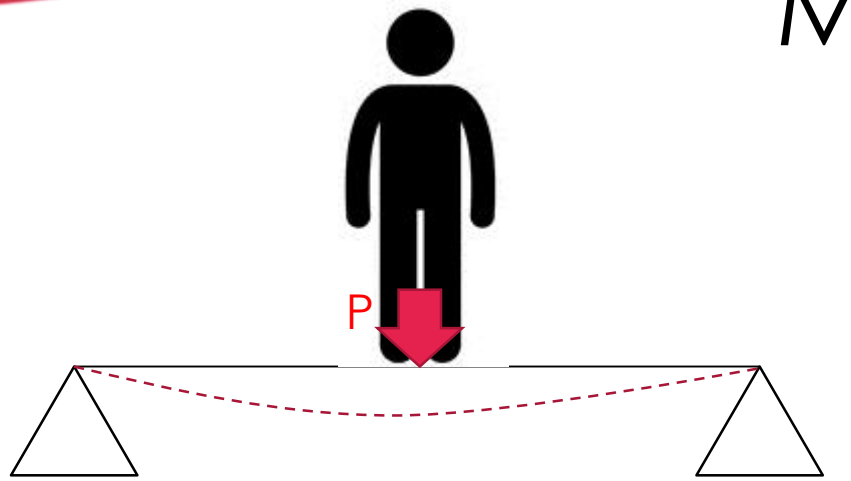
PURPOSE OF THE STUDY

- To design a software that automatically calculate the optimum thickness of core and skins of a composite sandwich, giving the material properties, the maximum stress applied to the sandwich and the desired stiffnes of the panel

- Material props.
- Face layup
- Desired stiffness
- Maximum load

- Optimum thickness of core and faces (that guarantees less weight)

MAXIMUM LOAD ESTIMATION



A skateboard may have either **STATIC LOAD** (the weight of the person) and **IMPULSE LOAD** (a jump down a step). The design must take care of the most critical condition.

STATIC LOAD

The weight of the person:

$$P=m*g$$

It is assumed a 90kg person, thus the static load is:

$$P=m*g=90*9.81=883N$$

IMPULSE LOAD

This load must be estimated under assumptions:

- The person «jumps down» a step (whose standard height is Δh=17cm)
- The wheels and supports of the skateboard are supposed as infinite-stiffness
- The damping factor of the material is supposed ξ= 0 (even if not realistic, it is a worst-case assumption)
- The maximum stress in the skateboard depends on the deformation admitted. In our case it is max_deform=15mm (in the center of the skateboard)

To calculate the impulse load, it is assumed that the person jumps, accumulates kinetic energy and the kinetic energy transforms to elastic energy that deformates the skateboard.

Under this, the impulse load is:

$$Potential\ energy: W_p = mg\Delta h = 150J = kinetic\ energy = elastic\ energy$$

$$= \frac{1}{2} * ke * \delta y^2 \rightarrow having\ imposed\ \delta y = 0{,}015m \rightarrow ke = \frac{2 * 150}{0{,}015^2} = 1{,}33\ e^6 N/m$$

$$MAXIMUM\ LOAD = ke * \delta y = 20000\ N$$

IMPULSE LOAD CONSIDERATIONS

The previous result is obviously a too-strong peak-force. This is because it was calculated assuming the human body as a rigid one, with infinite stiffness.

The reality is that the human body absorbs a quote of the kinetic energy, thus the one transformed in elastic energy is less than the previous calculated.

A setup was built, in order to measure the quote of energy absorbed by the body. Two persons (80kg and 50kg) jumped down a step, over a spring. The peak deformation of the spring corresponds to the peak force applied.

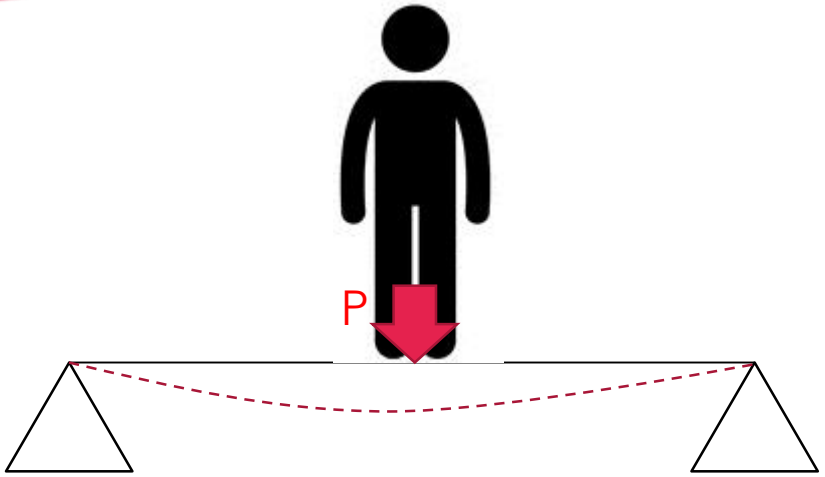
The result was that the average energy absorbed by the human body was 77% of the theorical for a completely-rigid body, yelding to:

$$Non - dissipated\ energy : W_p * 23\% = 150J * 0{,}23 = 34{,}5J = elastic\ energy$$

$$= \frac{1}{2} * ke * \delta y^2 \rightarrow having\ imposed\ \delta y = 0{,}015m \rightarrow ke = \frac{2 * 34{,}5}{0{,}015^2} = 0{,}31\ e^6 N/m$$

$$MAXIMUM\ LOAD = ke * \delta y = 4600\ N$$

Which is much less than that of a completely-rigid body



MAXIMUM LOADS

STATIC LOAD

P=883N

IMPULSE LOAD

P=4600N

The analisys follows using the worst-case: IMPULSE LOAD

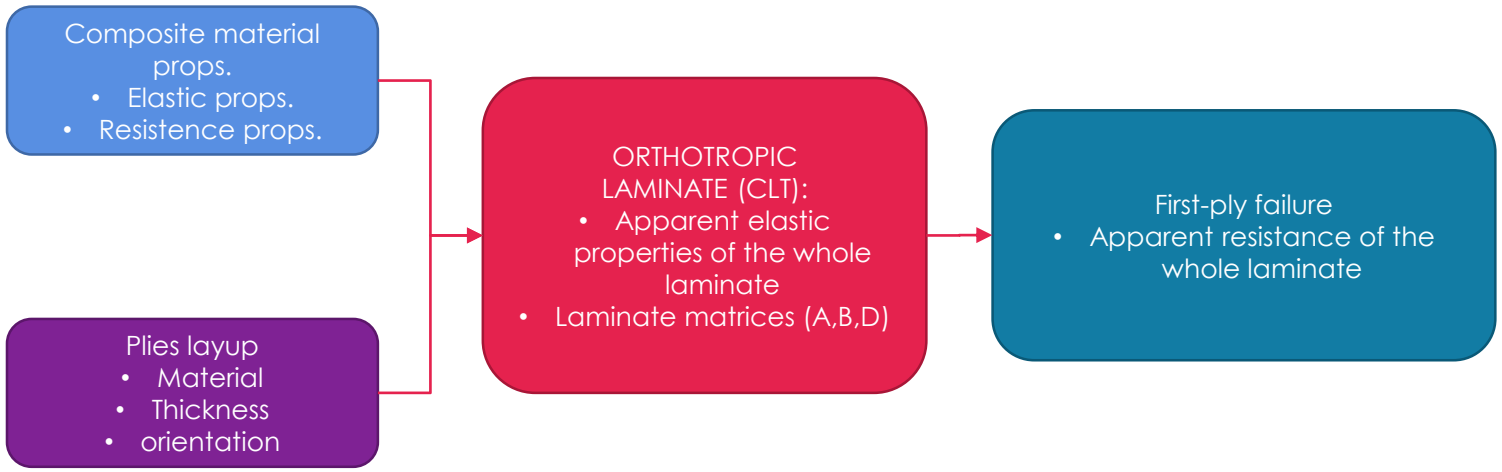
OPTIMIZATION SOFTWARE

The optimization software aims to satisfy the following 4 conditions:

- Guarantee a maximum deformation (in the center of the skateboard), thus a STIFFNESS CONSTRAINT
- Guarantee the avoidance of FACE FAILURE
- Guarantee the avoidance of CORE SHEAR
- Guarantee the avoidance of FACE WRINKLING

All this in the **less weigth** possible

SOFTWARE STRUCTURE




```
layup=[layer1;layer2;layer3;layer4];
laminate =  struct ('layup',layup,...
                    'A',[],...  always put an empty vector: it will be calculated after
                    'B',[],...  always put an empty vector: it will be calculated after
                    'D',[],...  always put an empty vector: it will be calculated after
                    'thick',0,...always put 0: it will be calculated after
                    'dens',0,... always put 0: it will be calculated after
                    'Exx',0,...  always put 0: it will be calculated after
                    'Eyy',0,...  always put 0: it will be calculated after
                    'Gxy',0,...  always put 0: it will be calculated after
                    'vxy',0,...  always put 0: it will be calculated after
                    'vyx',0);  % always put 0: it will be calculated after
```

Implementation of the classical lamination theory: this part is essential to derive the properties of the laminate:

- Elastic properties of the laminate (Ef)
- Resistance properties of the laminate (σf)

CLASSICAL LAMINATION THEORY

$$\boldsymbol{A_{ij}} = \sum_{k=1}^{n} \{Q_{ij}\}_n (z_k - z_{k-1})$$

$$\boldsymbol{B_{ij}} = \frac{1}{2} \sum_{k=1}^{n} \{Q_{ij}\}_n (z_k^2 - z_{k-1}^2)$$

$$\boldsymbol{D_{ij}} = \frac{1}{3} \sum_{k=1}^{n} \{Q_{ij}\}_n (z_k^3 - z_{k-1}^3)$$

```
function [ out ] = calculate_laminate_ABD( struct )
new_layup = struct.layup;

%calculate medium-plane:
tot_thick = 0;
tot_dens = 0;

for ind = 1:size(new_layup)
    h(ind) = tot_thick + new_layup(ind).thick; % this vector holds the vertical position of the interlaminate,referred to the top
    tot_thick = tot_thick + new_layup(ind).thick;
    tot_dens = tot_dens + new_layup(ind).material.dens*new_layup(ind).thick;
end

struct.thick=tot_thick;
struct.dens = tot_dens/tot_thick;
laminate_center=tot_thick/2;

A=zeros(3);
B=zeros(3);
D=zeros(3);

%Calculate the matrices
for ind = 1:size(new_layup)
    a = -laminate_center+h(ind);% z(k)
    if(ind==1)
        b = -laminate_center;% z(k-1)
    else
        b = -laminate_center+h(ind-1);
    end

    A = A+new_layup(ind).Q_hat*(a-b);
    B = B+1/2*new_layup(ind).Q_hat*(a^2-b^2);
    D = D+1/3*new_layup(ind).Q_hat*(a^3-b^3);

end

% combine the A,B,D matrices
K=[A B ; B D];
```

APPARENT PROPERTIES

$$\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix}; \quad \mathbf{K}^{-1} = \begin{bmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}} \\ \bar{\mathbf{B}} & \bar{\mathbf{D}} \end{bmatrix}$$

$$E_{x,app} = \frac{1}{H \cdot \bar{A}_{11}}, \quad E_{y,app} = \frac{1}{H \cdot \bar{A}_{22}}$$

$$\nu_{xy,app} = \frac{A_{12}}{A_{22}} = -\frac{\bar{A}_{12}}{\bar{A}_{11}}, \quad \nu_{yx,app} = \frac{A_{12}}{A_{11}} = -\frac{\bar{A}_{12}}{\bar{A}_{22}}$$

$$G_{xy,app} = G_{yx,app} = A_{66} = \frac{1}{H \cdot \bar{A}_{66}}$$

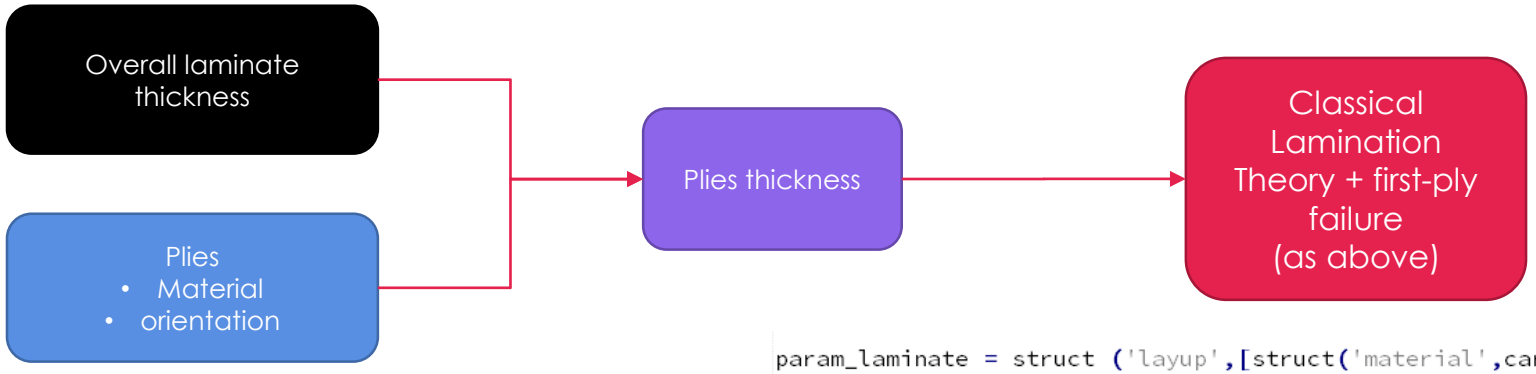
```
% combine the A,B,D matrices
K=[A B ; B D];
K_hat=inv(K);

% Apparent properties of the laminate

%correct expressions
Exx=1/(tot_thick*K_hat(1,1)); %[MPa] longitudinal elastic modulus
Eyy=1/(tot_thick*K_hat(2,2)); %[MPa] transversal elastic modulus
Gxy=1/(tot_thick*K_hat(3,3)); %[MPa] shear elastic modulus

%expressions used in CLT App
A_star=inv(A);
% Exx=1/(tot_thick*A_star(1,1)); %[MPa] longitudinal elastic modulus
% Eyy=1/(tot_thick*A_star(2,2)); %[MPa] transversal elastic modulus
% Gxy=1/(tot_thick*A_star(3,3)); %[MPa] shear elastic modulus

vxy = A(1,2)/A(2,2);
vyx = A(1,2)/A(1,1);
```

FIRST-PLY FAILURE

```
function [ output ] = calculate_first_ply_failure( struct )
% This function takes as input a laminate, then calculates the first-ply
% failure by applying the Tsai-Hill criterion. The hipotesys is to have
% only a longitudinal stress (Nx). The algorythm is explained at pag. 75 of
% the textbook.

% STEPS 1-2: determine Qhat, and A for the laminate (they were already
% calculated).
A= struct.laminate.A;

% auxiliary constants, as described at page 75 of the book
C3=(A(3,2)/A(3,3)*A(2,1)/A(2,2) - A(3,1)/A(3,3))/(1-A(3,2)/A(3,3)*A(2,3)/A(2,2));
C2=-A(2,3)/A(2,2)*C3 - A(2,1)/A(2,2);
C1=A(1,1)+A(1,2)*C2+A(1,3)*C3;

% STEP 3: apply unit-load Nx=1 and derive the epsilon_x_o
Nx=1;
epsilon_x_o = Nx/C1;

%now, for each lamina, calculate the main stresses
for k = 1:struct.num_layers
    norm_stresses = struct.laminate.layup(k).Q_mod * [1 0 0]';
    C1k(k)=norm_stresses(1);
    C2k(k)=norm_stresses(2);
    C3k(k)=norm_stresses(3);

    sigma1=C1k(k)*epsilon_x_o;
    sigma2=C2k(k)*epsilon_x_o;
    tau12=C3k(k)*epsilon_x_o;

    % STEP 4: by applying the Tsai-Hill criterion, determine the
    % most-stressed layer
    sigma_LU=struct.laminate.layup(k).material.sigma1;
    sigma_TU=struct.laminate.layup(k).material.sigma2;
    tau_LTU=struct.laminate.layup(k).material.tau12;
    H(k)=(sigma1/sigma_LU)^2-(sigma1/sigma_LU)*(sigma2/sigma_LU)+(sigma2/sigma_TU)^2+(tau12/tau_LTU)^2;
end

% find the maximum value of H, thus the first layer that will brake
[max_val,first_lay_brake] = max(H);

% now find the deformation that will bring to failure, by reversing the
% Tsai-Hill formula, and imposing H=1
epsilon_x_o_failure = sqrt(1/((C1k(first_lay_brake)/sigma_LU)^2-...
                              (C1k(first_lay_brake)/sigma_LU)*(C2k(first_lay_brake)/sigma_LU)+...
                              (C2k(first_lay_brake)/sigma_TU)^2+...
                              (C3k(first_lay_brake)/tau_LTU)^2));

% at this point it is easy to find the corresponding longitudinal stress:
Nx = C1*epsilon_x_o_failure;

struct.sigma_f=Nx;

output=struct;
end
```

It is an implementation of the iterative procedure described in the book

THE PARAMETRIC LAMINATE




```
param_laminate = struct ('layup',[struct('material',carbon,'angle',0*pi/180) ;...     layer1
                                  struct('material',carbon,'angle',90*pi/180) ;...    layer2
                                  struct('material',carbon,'angle',-90*pi/180) ;...   layer3
                                  struct('material',carbon,'angle',0*pi/180)] ,...    layer4
                         'laminate',laminate,... %it is needed a laminate struct
                         'num_layers',0,...  always put 0: it will be calculated after
                         'layer_thick',0,... %  always put 0: it will be calculated after
                         'sigma_f',0);%  [MPa] first-ply-failure always put 0: it will be calculated after
```

The limitation of the CLT is that the thickness of the laminate is the sum of the thicknesses of the plies, thus fixed. For the study of a sandwich it is necessary to impose the overall face thickness and to derive consequently the per-ply thickness.

In the parametric laminate the assumption is:

- All plies have the same thickness

Having this, it is possible to impose the parameter «overall laminate thickness» and have the ply thickness as the overall thickness/number of plies. The properties of the laminate are calculated as above, with the CLT and first-ply-failure

THE CALCULATION LOOP

In the calculation loop, there were defined:

- The core thickness interval (min-max=7-70mm)
- The face thickness interval (min-max=0.1-3mm)

It is a nested loop: the outer spans over core-thickness, the inner over face-thickness. For every couple of core-face thickness there were calculated:

- Panel stiffness
- Face failure maximum-load
- Core-shear maximum-load
- Face-wrinkling maximum-load

With all this data, for every step of the core thickness, it was calculated the minimum face thickness that accomplishes the load defined and the stiffness required.

```
for k=1:size(tc)

    for j=1:size(tf)
        % stiffness of the sandwich, as a function of core and face
        % thickness.
        % Thanks to semplification, the stiffness of the sandwich is equivalent to
        % that of the face, with respect to the sandwich's center:
        d=tf(j)+tc(k);
        D0(j,k)=((curr_laminate(j).laminate.Exx*b*1000*tf(j)*(d)^2)/2)*10^(-6); %([MPa*mm^4])e-6=([N*mm^2])e-6=[N*m^2]
        % specific weigth of the sandwich, as a function of core and face thickness
        sandw_dens(j,k)=((2*tf(j)*curr_laminate(j).laminate.dens) + resin.dens*tc(k))/(2*tf(j)+tc(k));%[kg/dm^3]

        % maximum load for FACE-FAILURE
        Pff(j,k)=1000*curr_laminate(j).sigma_f*b*tf(j)*d/(Bm*L);
        % maximum load for CORE-SHEAR
        Pcs(j,k)=1000*resin.tau * d / Bt;
        % maximum load for FACE-WRINKLING
        Pwr(j,k)=1000*tf(j)*d/(2*Bm*L)*(curr_laminate(j).laminate.Exx * resin.E * resin.G)^(1/3);

        % now define the functions that accomplishes the requests
        if((D0(j,k)>=EJ_min) && (tf(j)<min_stiff_face_thick(k)))
            min_stiff_face_thick(k)=tf(j);
        end
        if((Pff(j,k)>=P) && (tf(j)<min_Pff_face_thick(k)))
            min_Pff_face_thick(k)=tf(j);
        end
        if((Pcs(j,k)>=P) && (tf(j)<min_Pcs_face_thick(k)))
            min_Pcs_face_thick(k)=tf(j);
        end
        if((Pwr(j,k)>=P) && (tf(j)<min_Pwr_face_thick(k)))
            min_Pwr_face_thick(k)=tf(j);
        end
    end
```

Everyone of the face-failure, core shear, face wrinkling and stiffness, raised to a different minimum-face-thickness constraint. In order to guarantee all the failures and stiffness, the maximum of the minimum-thickness values was kept :

```
% now that the minimum face thickness is found, keep the worst
% condition and plot the corresponding density of the sandwich
min_face_thick(k) = max([min_stiff_face_thick(k),min_Pff_face_thick(k),min_Pcs_face_thick(k),min_Pwr_face_thick(k)]);

% having this data, derive the corresponding density:
panel_weigth(k)=(resin.dens * tc(k) + 2* min_face_thick(k) * curr_laminate(1).laminate.dens)*b*L/1000;
```

This determines the couple core-thickness/face-thickness that accomplishes the requirements.

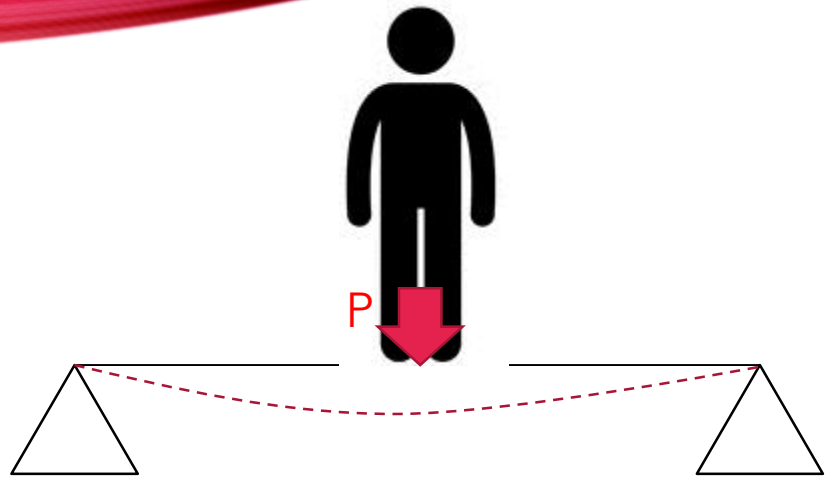
The 2 values, together with the skateboard dimensions, gives the overall skateboard weigth.

MINIMUM WEIGTH

The software calculates what is the minimum-weigth condition (that corresponds to a couple face-thickness/core-thickness), and plots the results in a graph

```
% Now find the minimum weigth and its index
[min_w index_min_w]=min(panel_weigth);

fig_title=sprintf('Minimum weigth comes when core is: %f mm, and face is: %f mm thick \n Total panel weigth is: %3f kg',tc(index_min_w),min_face_thick(index_min_w),panel_weigth(index_min_w));
plot(tc,[min_stiff_face_thick,min_Pff_face_thick,min_Pcs_face_thick,min_Pwr_face_thick,panel_weigth']);
title(fig_title);
legend('stiffness constraint','face failure','core shear','face wrinkling','panel density');
xlabel('core thickness [mm]');
ylabel('face thickness [mm]');
```

```
% stiffness of the sandwich, as a function of core and face
% thickness.
% Thanks to semplification, the stiffness of the sandwich is equivalent to
% that of the face, with respect to the sandwich's center:
d=tf(j)+tc(k);
D0(j,k)=((curr_laminate(j).laminate.Exx*b*1000*tf(j)*(d)^2)/2)*10^(-6); %([MPa*mm^4])e-6=([N*mm^2])e-6=[N*m^2]
```



STIFFNESS CONSTRAINT

In a situation like this (using the equation of elastic line), the deformation at the center is:

$$\eta_0 = \frac{P * L^3}{48 * EJ}$$

Thus it is possible to derive the term:

$$EJ = \frac{P * L^3}{48 * \eta_0}$$

This parameter has to be compared to the (simplified) stiffness expression for a sandwich panel:

$$D^0 = \frac{E^f b t^f d^2}{2} \geq EJ$$

The stiffness of the panel must be greater than EJ

```
% maximum load for FACE-FAILURE
Pff(j,k)=1000*curr_laminate(j).sigma_f*b*tf(j)*d/(Bm*L);
% maximum load for CORE-SHEAR
Pcs(j,k)=1000*resin.tau * d / Bt;
% maximum load for FACE-WRINKLING
Pwr(j,k)=1000*tf(j)*d/(2*Bm*L)*(curr_laminate(j).laminate.Exx * resin.E * resin.G)^(1/3);
```

FAILURES

The maximum load possible depends on the type of failure it is intended to avoid:

- Guarantee the avoidance of FACE FAILURE

$$P_{ff} = \frac{\sigma_{f,max} b t^f d}{B_M L}$$

- Guarantee the avoidance of CORE SHEAR

$$P_{cs} = \frac{\tau_{c,max} d}{B_T}$$

- Guarantee the avoidance of FACE WRINKLING

$$P_{fw} = \frac{t^f d}{2 B_M L} \sqrt[3]{E_f E_c G_c}$$

NOTES: the tensile strength of the faces is that of the first-ply-failure for a longitudinal stress.

In the previous formulas, due to load conditions Bm=1/4, Bt=1/2.

MATERIALS

For the materials, the choice was to evaluate 2 different materials for faces and 2 different materials for core:

FACES:

Unidirectional carbon:

```
% carbon unidirectional
carbon = struct('E11',135000,...    %[MPa] tensile modulus
                'E22',10000,...     %[MPa]
                'G12',5000,...      %[MPa]
                'v12',0.3,...
                'sigma1',1500,...   %[MPa] tensile strength
                'sigma2',50,...     %[MPa]
                'tau12',70,...      %[MPa] shear strength
                'dens',1.6,...      %[kg/dm^3]
```

Fabric carbon:

```
% % carbon fabric
carbon = struct('E11',75000,...     %[MPa] tensile modulus
                'E22',75000,...     %[MPa]
                'G12',5000,...      %[MPa]
                'v12',0.1,...
                'sigma1',600,...    %[MPa] tensile strength
                'sigma2',600,...    %[MPa]
                'tau12',90,...      %[MPa] shear strength
                'dens',1.6,...      %[kg/dm^3]
```

CORE:

Low-cost poliurethane foam:

```
% % poliurethane foam
resin = struct('E',25.7,...      %[MPa] tensile modulus
               'v',0.3,...
               'G',3.93,...    %Gr= Er/(2*(1+vr)) shear modulus
               'sigma',0.8,... %[MPa] tensile strength
               'tau',0.441,... %[MPa] shear strength
               'dens',0.096);  %[kg/dm^3]
```

High-quality PVC foam:

```
% high-quality PVC foam
resin = struct('E',95,...      %[MPa] tensile modulus
               'v',0.7,...
               'G',27,...    %Gr= Er/(2*(1+vr)) shear modulus
               'sigma',2.5,... %[MPa] tensile strength
               'tau',1.15,... %[MPa] shear strength
               'dens',0.08);  %[kg/dm^3]
```

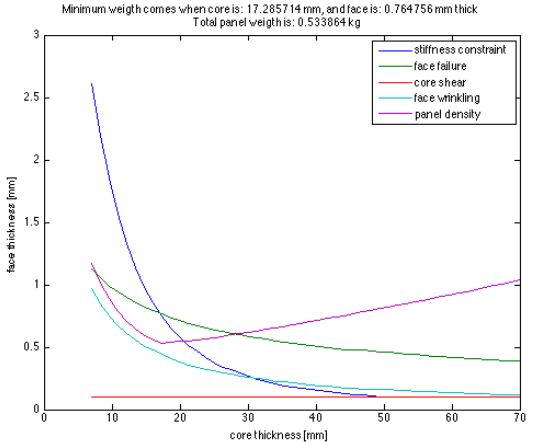
LAYUP

Even if the software can evaluate every type of layup, a very simple layup was chosen:

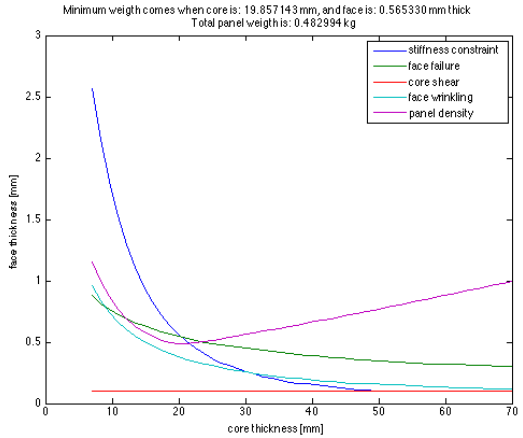
```
param_laminate = struct ('layup',[struct('material',carbon,'angle',0*pi/180) ;...     layer1
                                  struct('material',carbon,'angle',90*pi/180) ;...    layer2
                                  struct('material',carbon,'angle',-90*pi/180) ;...   layer3
                                  struct('material',carbon,'angle',0*pi/180)] ,...    layer4
```

RESULTS

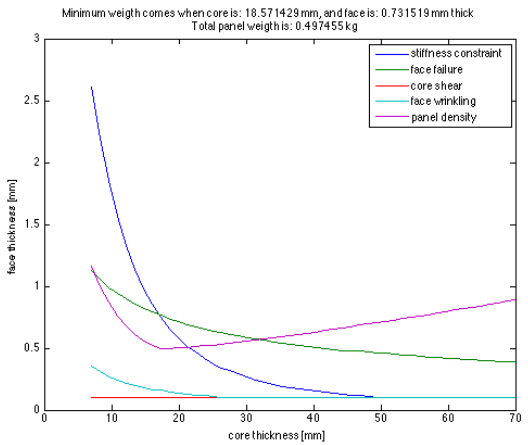
The previous materials were combined, yelding to 4 combinations:



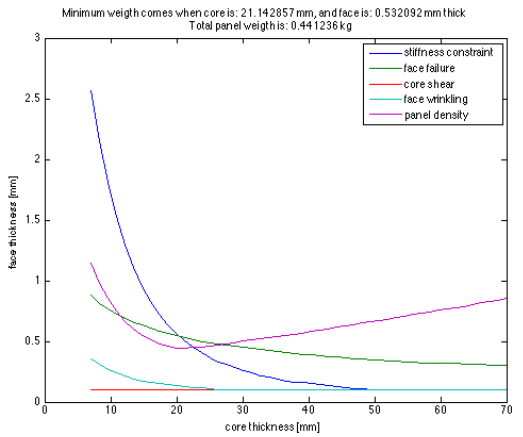
UD + poliurethane



fabric + poliurethane



UD + PVC



fabric + PVC

F.E.M. SIMULATIONS

In order to validate the previous work, a FEM analisys was done.

The problem was simplified by using a single panel made by a «shell», where there were introduced all the layers composing the sandwich (4 plier per skin + the core).

The materials combination chosen for the simulation was «carbon-fabric+PVC».

The table was vinculated at the 2 extremes and a distributed load of 4600N was applied in the central line.

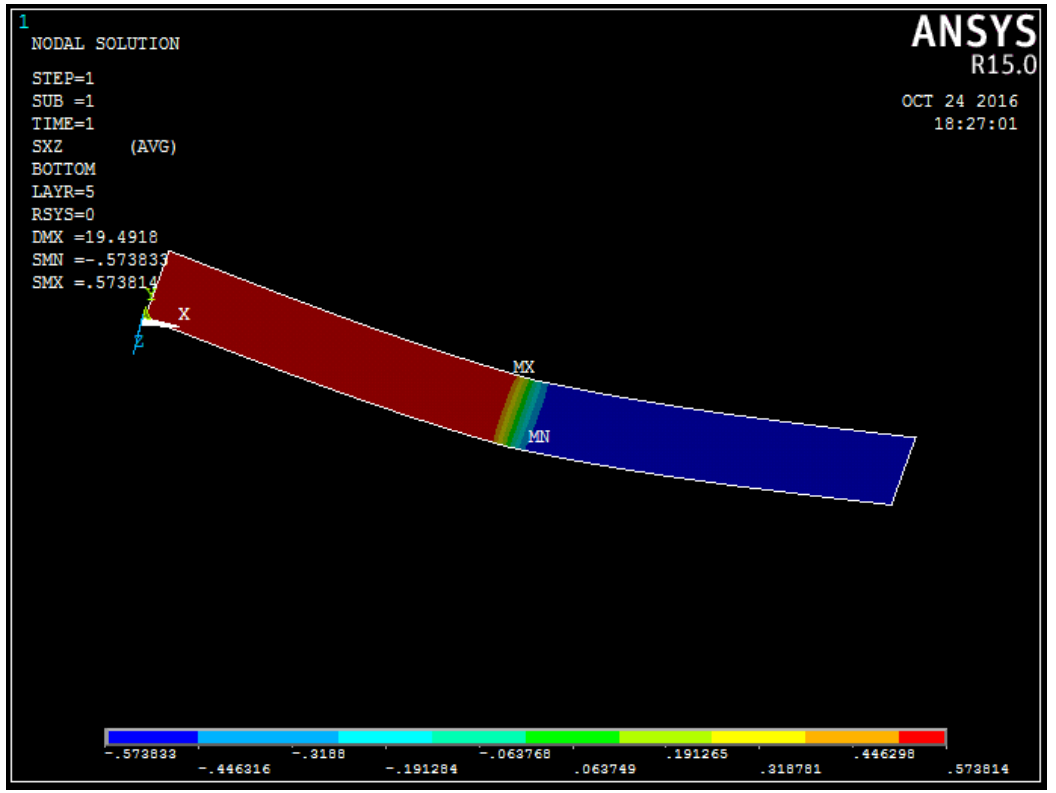
In the following sheets, the results:

DEFORMATION AND CORE SHEAR

As visible, the maximum core shear stress is less than the limit of the material (1.15 Mpa).

The deformation in the center is more than that expected (and designed for: 15mm, against the 19 found).

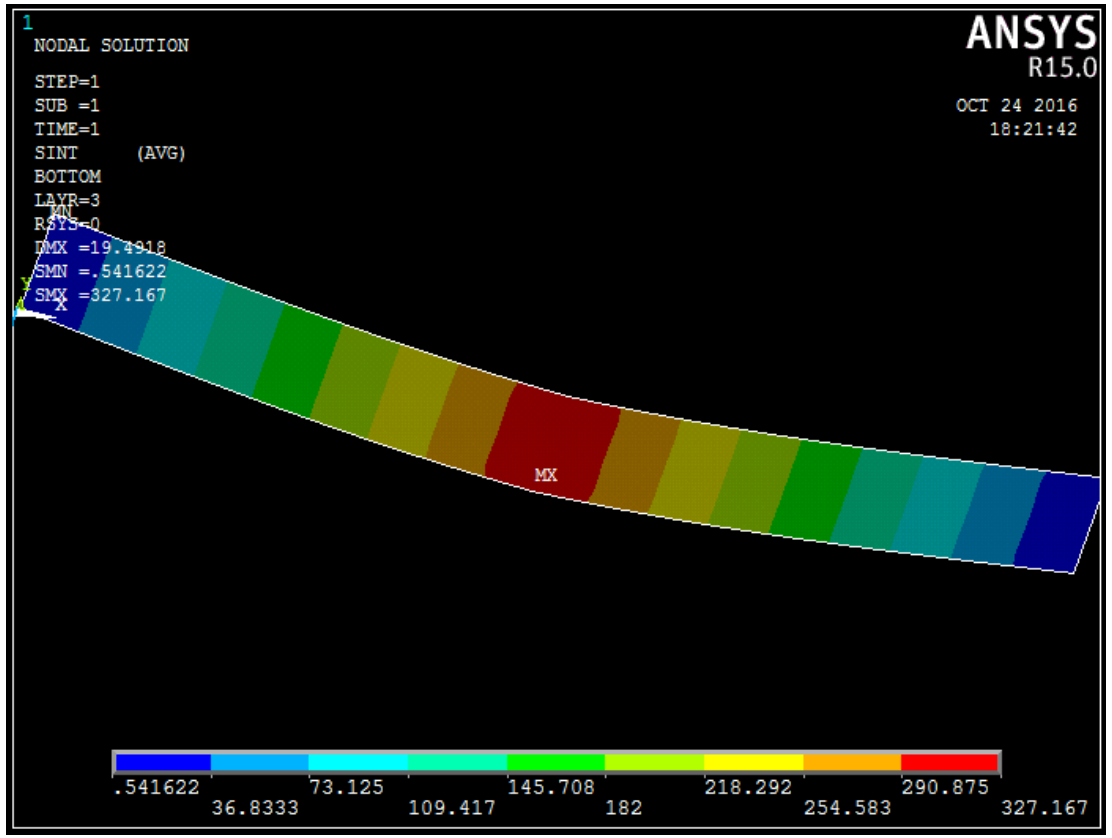
Seamlessly, the simplification in the calculus has led to an error.



FACE STRESS

The maximum stress is less than thato supported by the material (approx. 550-600MPa), so the design is SAFE.

The peak stress absolute value is very similar in all the layers (coherent with the type of the composite ply: fabric)



CONSIDERATIONS

The present work aimed to find the conditions to build a shateboard that is as-light as possible (giving the materials and layup).

It was found that it is a very powerful tool, that also gave the way to determine that, in the space of the materials chosen, the less-weigth one is that with **FABRIC+PVC foam**.

In this last condition, a FEM simulation was performed. It demonstrated that the design is safe, because **the maximum stress conditions are observed**.

The simulation demonstrated that **the maximum panel deformation was greater than that required** (approx. +25%). Probably this was due to simplification in calculus and in simulation.

This design requirement was not that «strict», but if it s important to improve the design procedure, a less-simplified approach must be kept.

FUTURE WORK

The current software only calculates the stresses and first-ply-failure for a LONGITUDINAL LOAD, it is possible to extend it for other load types.

It is possible to extend the functionality of the software by adding a «safety cohefficient», in order to design the panel with some «margins».

THANK YOU FOR THE ATTENTION