

KUNGLIGA TEKNISKA HÖGSKOLAN



MSc Vehicle Engineering

A.Y. 2024/25

**Structural Optimisation and Sandwich Design
Design Report**

Carlo Vittorio Colucci: cvco@kth.se

January 13, 2025

Design project introduction

This project is split into 4 main tasks. The main topic in common is the optimization problem-solving.

The *1st* task is related to the understanding of the optimization problem itself, applying it to the simple study case of a beam subjected to a distributed load.

The *2nd* task exploits the optimization technique to retrieve the material mechanical properties of a sandwich beam.

The *3rd* task deals with the analysis of the behavior of a sandwich panel, when subjected to a certain load, according to the DNV rules.

The *4th* task consists of designing a panel of the hull of the Swedish Coast Guard vessel KBV 312.

Contents

| | | |
|----------|--|-----------|
| 1 | Task 1A - Unconstrained optimization | 1 |
| 1.1 | Theory | 1 |
| 1.2 | Method | 1 |
| 1.3 | Results | 2 |
| 1.4 | Discussion | 4 |
| 2 | Task 1B - Equilibrium displacement shape determination of a sandwich beam | 5 |
| 2.1 | Theory | 5 |
| 2.2 | Method | 6 |
| 2.3 | Results | 9 |
| 2.4 | Discussion | 9 |
| 3 | Task 2 - Experimental determination of material properties | 11 |
| 3.1 | Theory | 11 |
| 3.2 | Experimental setup | 12 |
| 3.3 | Method | 13 |
| 3.4 | Results | 16 |
| 3.5 | Discussion | 16 |
| 4 | Task 3 - Sandwich plate analysis | 17 |
| 4.1 | Theory | 17 |
| 4.2 | Experimental setup | 17 |
| 4.3 | Method | 18 |
| 4.4 | Results | 18 |
| 4.5 | Discussion | 19 |
| 5 | Task 4 - Sandwich hull design | 20 |
| 6 | Task 4.1 - Simple panel optimization | 21 |
| 6.1 | Theory | 21 |
| 6.2 | Experimental setup | 21 |
| 6.3 | Method | 21 |
| 6.4 | Results | 23 |
| 6.5 | Discussion | 25 |

| | | |
|----------|--|-----------|
| 7 | Task 4.2 - Panel with stiffener optimization (basic) | 26 |
| 7.1 | Theory | 26 |
| 7.2 | Experimental setup | 27 |
| 7.3 | Method | 28 |
| 7.4 | Results | 29 |
| 7.5 | Discussion | 30 |
| 8 | Task 4.3 - Panel with stiffener optimization (extended) | 31 |
| 8.1 | Theory | 31 |
| 8.2 | Method | 31 |
| 8.3 | Results | 33 |
| 8.4 | Discussion | 34 |
| 9 | Task 4.4 - Sandwich hull design | 35 |
| 9.1 | Experimental setup | 35 |
| 9.2 | Method | 35 |
| 9.3 | Results | 36 |
| 9.4 | Discussion | 36 |

1 Task 1A - Unconstrained optimization

The goal of Task 1A is to develop an algorithm to solve an optimization problem: given a set of initial states, find the optimal set minimizing a defined cost function.

A sensitivity analysis of the algorithm is carried out, for a better understanding of its working principle and its main points of weakness.

1.1 Theory

This task is undertaken by means of the Newton's method implementation.

The starting point is to define an approximation of the cost function for a given combination of states:

$$\hat{f} = f(x^k) + f'(x^k) \cdot (x - x^k) + \frac{1}{2} \cdot (x^k) \cdot (x - x^k)^2 \quad (1)$$

To find the minimum of this function $\min \hat{f}(x)$, an initial guess of states combination is chosen and the condition $\hat{f}'(x) = 0$ is imposed, leading to define a next states combination $x^{k+1} = x_k - \frac{f'(x^k)}{f''(x^k)}$ closer to the minimum. This procedure is effective for functions depending from a single state, but the optimization problem could depend on a higher number of states. Thus, a more generic method is defined.

1.2 Method

The main steps to follow are:

1. Guessing a combination of states x^k as starting point;
2. Computing the next states combination closer to the minimum: $x^{k+1} = x^k - \frac{\nabla f(x^k)^T}{H(x^k)}$, where $\nabla f(x^k)^T$ and $H(x^k)$ are the function Hessian matrix and gradient at the current states combination;
3. Check if the step-length is lower than a fixed tolerance: $l = |x^{k+1} - x^k| < \epsilon$;
4. Return at point 2 if the step-length is too high, otherwise the states minimizing the cost function are found: $x^{k+1} = x^*$.

A further check, to verify that the minimum is found, is to verify if the first and second optimality conditions are met: $\nabla f(x^*) < \epsilon, H(x^*) < \epsilon$.

The code section related to this method follows:

```
1 x = x0;
2 steplength = 1;
3 i = 0;
4 xIterates = [x0];
5
6 while((steplength > ftol) & i < 100)
7     i = i + 1 ;
8     gr = gradf(x);
9     hf = hessf(x);
10    x1 = x - hf\gr';
11    steplength = abs(x1-x);
12    x = x1;
13    xIterates = [xIterates, x1];
14 end
15
16 if gradf(x)<ftol
17     disp('First_order_optimality_condition_satisfied.')
18 else
19     disp('First_order_optimality_condition_not_satisfied.')
20 end
21
22 if hessf(x)>0
23     disp('Second_order_optimality_condition_satisfied.')
24 else
25     disp('Second_order_optimality_condition_not_satisfied.')
26 end
```

1.3 Results

The cost function used for this task is:

$$f(x) = (x_1 - 1)^4 + 2 \cdot x_1 \cdot x_2 + 8 \cdot x_2^2, \quad \text{where } x = [x_1, x_2]^T \quad (2)$$

Considering as initial guess: $x^0 = [2.0, 1.5]$, the number of iterations required to reach the $f(x)$ minimum is 5. The states corresponding to the minimum are $x^* = [1.45, -0.18]$, and the cost function value is $f(x^*) = -0.22$. The iterate values are reported in the following table:

| Iteration | x_1 | x_2 |
|-----------|-------|-------|
| 0 | 2.0 | 1.5 |
| 1 | 1.70 | -0.21 |
| 2 | 1.53 | -0.19 |
| 3 | 1.46 | -0.18 |
| 4 | 1.45 | -0.18 |
| 5 | 1.45 | -0.18 |

Table 1: Iterates values for $x^0 = [2.0, 1.5]$

The Matlab code reported in the previous Section 1.2 returned fulfilled both the first and the second order optimality conditions.

It is interesting to observe the results of starting from another combination of states, such as $x^0 = [1.05, 1.5]$. In this case, the number of iterations increases to 12, but the states corresponding to the minimum are again $x^* = [1.45, -0.18]$. The iterates values are reported in the following table:

| Iteration | x_1 | x_2 |
|-----------|-------|-------|
| 0 | 1.05 | 1.5 |
| 1 | -0.14 | 0.02 |
| 2 | 0.24 | -0.03 |
| 3 | 0.51 | -0.06 |
| 4 | 0.74 | -0.09 |
| 5 | 1.20 | -0.15 |
| 6 | 2.29 | -0.29 |
| 7 | 1.88 | -0.24 |
| 8 | 1.63 | -0.20 |
| 9 | 1.50 | -0.19 |
| 10 | 1.45 | -0.18 |
| 11 | 1.45 | -0.18 |
| 12 | 1.45 | -0.18 |

Table 2: Iterates values for $x^0 = [1.05, 1.5]$

For a proper sensitivity analysis, other tests are performed for different combinations of

initial states: $x_1^0 \in [-1.5, 3.5]$; $x_2^0 \in [-2.5, 3.0]$. All combinations lead to the same final set of states $x^* = [1.45, -0.18]$. The maximum number of iterations is 24. The trajectories are reported here:

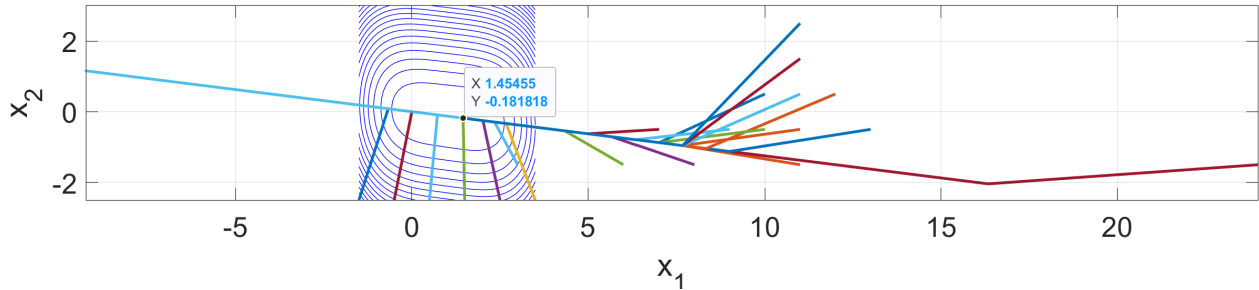


Figure 1: Minimization trajectories

1.4 Discussion

To undertake a more effective discussion of the results, it is helpful to graphically represent the cost function:

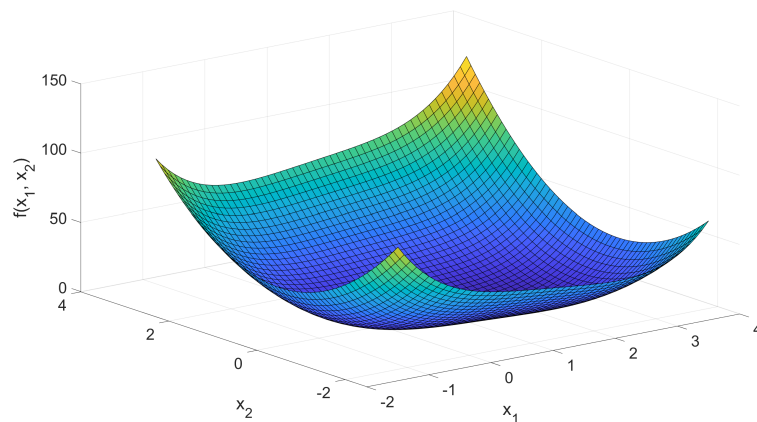


Figure 2: Cost function surface

The cost function looks to be convex. This characteristic is confirmed by the fact that each starting states set always leads to the same final states set corresponding to the minimum. Furthermore, the number of iterations the algorithm needs to converge is always low, indicating good performance.

In Figure 1 some trajectories are outside the cost function starting domain. This is justified by some regions of this function, which are characterized by a negative curvature, leading to a very high step-length.

2 Task 1B - Equilibrium displacement shape determination of a sandwich beam

The goal of Task 1B is to find the polynomial coefficients of the equations describing the equilibrium displacement shape of a sandwich beam. The load case is a distributed load and the support is a one side clamp. The setup is depicted here:

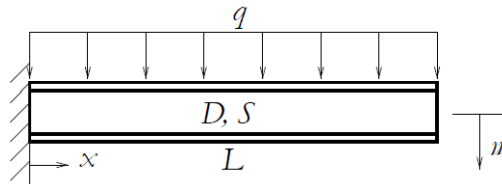


Figure 3: Task 1B setup

The sandwich beam characteristics and load are reported in the following table:

| Parameter | Value | Unit measure |
|-----------|-------------------|--------------|
| D | $96.1 \cdot 10^6$ | $N \cdot mm$ |
| S | 961 | N/mm |
| L | 500 | mm |
| q | 10 | N/mm |

Table 3: Sandwich beam characteristics

2.1 Theory

The equations describing the displacement shape (w_b, w_s) are strictly related to the total work on the beam (Π), which is made up of two contributions:

1. The elastic strain energy in the beam:

$$U_1 = \frac{1}{2} \cdot \int_0^L D \cdot \left(\frac{\partial^2 w_b}{\partial x^2} \right)^2 + S \cdot \left(\frac{\partial w_s}{\partial x} \right)^2 dx \quad (3)$$

2. The potential energy of the external loads:

$$U_1 = - \int_0^L q \cdot w dx \quad (4)$$

Leading to:

$$\Pi = U_1 + U_2 \quad (5)$$

The shape equations must be coherent with the proportion their differentiation orders have with the transverse force and the bending moment. Furthermore, the kinematic boundary conditions must be fulfilled. In the end, they are described by means of the following polynomial functions:

$$w_s = a_1 \cdot x + a_2 \cdot x^2, \quad w_b = b_2 \cdot x^2 + b_3 \cdot x^3 + b_4 \cdot x^4 \quad (6)$$

2.2 Method

To retrieve the polynomial coefficients characterizing Equations 6, an optimization problem is solved. The cost function is the total work, depending on the states values $c = [a_1, a_2, b_2, b_3, b_4]$. The goal is to retrieve the set of c corresponding to the minimum of the cost function: $c^* = \text{argmin}(\Pi(c))$.

Two methods are used to solve this optimization problem: the Newton's method introduced in Section 1.2 and the Matlab integrated function `fminsearch`.

The termination criterion used for the Netwon's method is the step-length magnitude check to be lower than a fixed tolerance.

The code section related to this part follows:

```

1 syms a1 a2 b2 b3 b4 x real
2 syms D S q L real
3 coefficients = [a1 a2 b2 b3 b4];
4 constants = [D S q L];
5
6 ws = a1*x + a2*x^2;
7 wb = b2*x^2 + b3*x^3 + b4*x^4;
8
9 ddwb = diff(diff(wb,x),x);
10 dws = diff(ws,x);
11
12 w = ws + wb;
13
14 U1 = D*ddwb^2 + S*dws^2;
15 U1 = int(U1, x, 0, L);

```

```
16 U1 = 0.5 * U1;
17
18 U2 = q*w;
19 U2 = -int(U2, x, 0, L);
20
21 PI = U1+U2
22
23 gradPI = [diff(PI,a1), diff(PI,a2), diff(PI,b2), diff(PI,b3),
           diff(PI,b4)];
24 hessianPI = [diff(diff(PI,a1),a1) diff(diff(PI,a1),a2)
              diff(diff(PI,a1),b2) diff(diff(PI,a1),b3) diff(diff(PI,a1),b4)
              diff(diff(PI,a2),a1) diff(diff(PI,a2),a2) diff(diff(PI,a2),b2)
              diff(diff(PI,a2),b3) diff(diff(PI,a2),b4)
              diff(diff(PI,b2),a1) diff(diff(PI,b2),a2) diff(diff(PI,b2),b2)
              diff(diff(PI,b2),b3) diff(diff(PI,b2),b4)
              diff(diff(PI,b3),a1) diff(diff(PI,b3),a2) diff(diff(PI,b3),b2)
              diff(diff(PI,b3),b3) diff(diff(PI,b3),b4)
              diff(diff(PI,b4),a1) diff(diff(PI,b4),a2) diff(diff(PI,b4),b2)
              diff(diff(PI,b4),b3) diff(diff(PI,b4),b4)];
29
30 gradPI = subs(gradPI, constants, parameters);
31 hessianPI = subs(hessianPI, constants, parameters);
32 hessianPI = double(hessianPI);
33
34 ftol = 0.0001;
35 x0 = ones(5,1);
36 x = x0;
37 steplength = 1;
38 i = 0;
39 xIterates = x0;
40 while((steplength > ftol) & i < 100)
41     i = i + 1;
42     gr = subs(gradPI, coefficients, x');
43     gr = double(gr);
44     x1 = x - hessianPI\gr';
45     steplength = abs(x1-x);
46     x = x1;
```

```

47     xIterates = [xIterates, x1];
48 end
49
50 a1real = qpar*Lpar/Spar;
51 a2real = -qpar/Spar/2;
52 b2real = qpar*Lpar^4/(24*Dpar)*6/Lpar^2;
53 b3real = -qpar*Lpar^4/(24*Dpar)*4/Lpar^3;
54 b4real = qpar*Lpar^4/(24*Dpar)/(Lpar^4);
55
56 realValues = [a1real; a2real; b2real; b3real; b4real]
57 if abs(realValues-x)<ftol
58     disp('Solution□verified.□')
59 else
60     disp('Solution□not□verified.□')
61 end
62
63 PI = subs(PI, constants, parameters);
64 PIcoeff = flip(double(coeffs(PI)));
65 fun = @(x) (PIcoeff(1)*x(1).^2 + PIcoeff(2)*x(1).*x(2) +
66     PIcoeff(3)*x(1) + ...
67     PIcoeff(4)*x(2).^2 + PIcoeff(5)*x(2) + PIcoeff(6)*x(3).^2 + ...
68     PIcoeff(7)*x(3).*x(4) + PIcoeff(8)*x(3).*x(5) + PIcoeff(9)*x(3) + ...
69     PIcoeff(10)*x(4).^2 + PIcoeff(11)*x(4).*x(5) + PIcoeff(12)*x(4) + ...
70     PIcoeff(13)*x(5).^2 + PIcoeff(14)*x(5));
71 x0 = zeros(5,1);
72 options = optimset('MaxFunEvals', 1e4, 'MaxIter', 1e4); % Increase
73     limits
74 [xMinSearch,fval,exitflag,output] = fminsearch(fun, x0',options);

```

In this Section the use of symbolic variables is crucial to ease the process and to avoid errors due to hand made computations.

For the Netwon's method, the symbolic variables are used during the whole procedure up to the computation of the cost function gradient and hessian matrix for a given set of states.

Instead, for the `fminsearch` method, the symbolic variables are used to evaluate the coefficients of the total work equation, that is directly fed to the Matlab function, with an initial states combination and the function settings.

The initial states used are $x_0 = [0 \ 0 \ 0 \ 0 \ 0]^T$. For different combinations, it is observed that

the function doesn't converge to the correct result.

2.3 Results

The real polynomial coefficients values are known from the outset and are used to verify the correctness of the computed ones. Their values are:

| Coefficient | Value |
|-------------|------------------------|
| a_1 | 5.203 |
| a_2 | $-5.203 \cdot 10^{-3}$ |
| b_2 | $6.504 \cdot 10^{-3}$ |
| b_3 | $-8.672 \cdot 10^{-6}$ |
| b_4 | $4.336 \cdot 10^{-9}$ |

Table 4: Polynomial coefficients values

The results from the Newton's method and the `fminsearch` function both converge to the real values.

The major difference between the two is the number of iterations required to converge: 2 for the Newton's method and 468 for the `fminsearch` function.

2.4 Discussion

The Newton's method takes slightly more time and requires a basic theory knowledge to be set up. Nevertheless, its performance is much higher than the `fminsearch` function, even if the goal is reached with both of them.

The reason why `fminsearch` requires more iterations is related to the algorithm behind: it is a direct search method (based on attempts) and it doesn't use analytical gradients, as in the Newton's method. An illustration of how the points are computed at each iteration is shown:

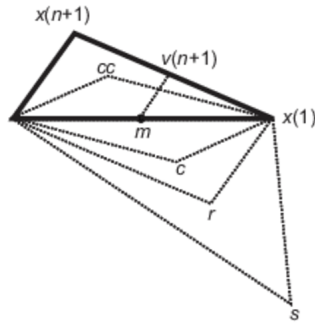


Figure 4: Iteration process in `fminsearch`

3 Task 2 - Experimental determination of material properties

The goal of Task 2 is to retrieve the material properties of a sandwich beam by analyzing its frequency response during a vibration test.

The properties of interest are the face bending stiffness (E_F) and the core shear stiffness (G_C).

3.1 Theory

The displacement of a sandwich beam is generically described by the following equation:

$$w(x, t) = \phi(x) \cdot \psi(t) \quad (7)$$

where:

$$\begin{aligned} \phi(x) &= C_1 \cdot \sinh(\alpha \cdot x) + C_2 \cdot \cosh(\alpha \cdot x) + C_3 \cdot \sin(\beta \cdot x) + C_4 \cdot \cos(\beta \cdot x) \\ \psi(t) &= \sin(\omega \cdot t) \end{aligned} \quad (8)$$

The boundary conditions (BC) for this experiment (detailed explanation in Section 3.2) correspond to the ones of a free-free beam:

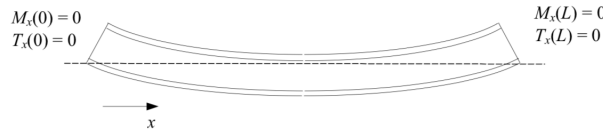


Figure 5: Boundary conditions for a free-free beam

That are formalized in the following way:

$$\begin{aligned} \frac{\partial^2 w}{\partial x^2} \Big|_{x=0,L} + \frac{\rho \cdot \omega^2}{S} \cdot w \Big|_{x=0,L} &= 0 \\ D \cdot \frac{\partial^3 w}{\partial x^3} \Big|_{x=0,L} + \frac{D \cdot \rho \cdot \omega^2}{S} \cdot \frac{\partial w}{\partial x} \Big|_{x=0,L} &= 0 \end{aligned}$$

These BCs, when considering Equation 8, lead to the following equation:

$$A \cdot [C_1, C_2, C_3, C_4]^T = 0 \quad (9)$$

where $A \in \mathbb{R}^{4 \times 4}$ depends on the beam dimensions and its materials properties: $A \propto D, S, L, \rho, \omega$:

$$A = \begin{bmatrix} 0 & S_{11} & 0 & S_{22} \\ \alpha \cdot S_{11} & 0 & \beta \cdot S_{22} & 0 \\ S_{11} \cdot \sinh(\alpha \cdot L) & S_{11} \cdot \cosh(\alpha \cdot L) & S_{22} \cdot \sin(\beta \cdot L) & S_{22} \cdot \cos(\beta \cdot L) \\ \alpha \cdot S_{11} \cdot \cosh(\alpha \cdot L) & \alpha \cdot S_{11} \cdot \sinh(\alpha \cdot L) & \beta \cdot S_{22} \cdot \cos(\beta \cdot L) & -\beta \cdot S_{22} \cdot \sin(\beta \cdot L) \end{bmatrix}$$

with:

$$\alpha = \sqrt{-\frac{\rho \cdot \omega^2}{2 \cdot S} + \sqrt{\left(\frac{\rho \cdot \omega^2}{2 \cdot S}\right)^2 + \frac{\rho \cdot \omega^2}{D}}};$$

$$\beta = \sqrt{\frac{\rho \cdot \omega^2}{2 \cdot S} + \sqrt{\left(\frac{\rho \cdot \omega^2}{2 \cdot S}\right)^2 + \frac{\rho \cdot \omega^2}{D}}};$$

$$S_{11} = \omega^2 \cdot \rho + S \cdot \alpha^2;$$

$$S_{22} = \omega^2 \cdot \rho - S \cdot \beta^2.$$

Equation 9 has a solution, apart of the trivial one $[C_1, C_2, C_3, C_4]^T = 0$, for:

$$\omega_i : \det(A(\omega)) = 0 \quad (10)$$

The frequency values solving this equation correspond to the beam natural frequencies.

3.2 Experimental setup

The experimental setup used for this testing is shown in the following picture:

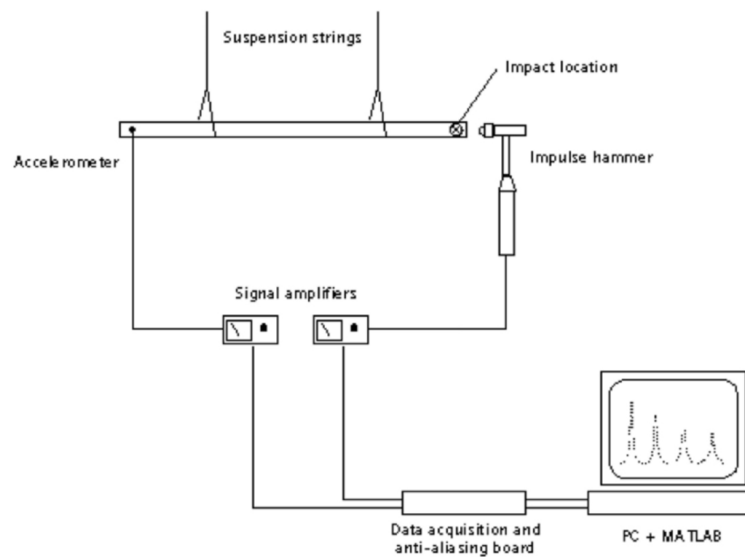


Figure 6: Experimental setup

The impulse hammer is used to hit the beam. During the collision phase, the generated force is measured to understand if the hit is effective and reliable or not.

The accelerometer is placed on the beam tips, in order to avoid that, during the structure excitation, the deformation nodes positions correspond to the accelerometer one. This would lead to an not effective measurement.

A PC with a preset Matlab script is used to collect the data coming from both the impulse hammer and the accelerometer. At the end of each test, the script directly returns the frequency response function.

The sandwich beam used for the test is made up of carbon fiber faces and a foam typical for sandwich structures applications. Its main properties are reported in the following table:

| Parameter | Value | Unit measure |
|----------------|----------------------|--------------|
| Core thickness | $25.4 \cdot 10^{-3}$ | m |
| Face thickness | $0.72 \cdot 10^{-3}$ | m |
| Beam length | $852 \cdot 10^{-3}$ | m |
| Beam width | $40 \cdot 10^{-3}$ | m |
| Beam area | 0.2083 | kg |

Table 5: Sandwich beam properties

3.3 Method

The experimental procedure is really simple. It consists of:

- Starting the data collection phase by the preset Matlab script;
- Ensuring that there are no interferences between the experimental equipment and possible external objects;
- Gently hitting the beam by the impulse hammer. The idea is that an impulse input excites the sandwich beam at all its frequencies. As known from theory, only in correspondence of the beam natural frequencies the acceleration magnitude is amplified;
- Waiting a preset time for a correct data collection, without feeding the impulse hammer with any other possible external input;
- Processing the test collected results.

```

1 function eigF = eigFresearch(parameters)
2     L = 852e-3; % [m]
3     M = 0.2083; % [kg]
4     a = 40e-3; % [m] width
5     rho = M/(L*a); % [kg/m^2]
6     function detA = detAcomputation(w)
7         alpha = sqrt( -rho*w.^2/(2*parameters(2)) +
8             sqrt((rho*w.^2/(2*parameters(2)))^2+rho*w.^2/parameters(1))
9             );
10        beta = sqrt( rho*w.^2/(2*parameters(2)) +
11            sqrt((rho*w.^2/(2*parameters(2)))^2+rho*w.^2/parameters(1))
12            );
13        S11 = w.^2*rho + parameters(2)*alpha.^2;
14        S22 = w.^2*rho - parameters(2)*beta.^2;
15
16        A = [0 S11 0 S22
17            alpha.*S11 0 beta.*S22 0
18            S11.*sinh(alpha*L) S11.*cosh(alpha*L) S22.*sin(beta*L)
19            S22.*cos(beta*L)
20            alpha.*S11.*cosh(alpha*L) alpha.*S11.*sinh(alpha*L)
21            beta.*S22.*cos(beta*L) -beta.*S22.*sin(beta*L)];
22        detA = det(A);
23    end
24    w0 = 0.5 * (22.4*sqrt(parameters(1)/(rho*L^4)));
25    eigF = fzero(@detAcomputation,w0);
26    eigF = [eigF, fzero(@detAcomputation,2*eigF)];
27 end
28
29 function J = costFunction(parameters)
30     eigF = eigFresearch(parameters);
31     J = (eigF(1)-1248.5)^2 + (eigF(2)-3077.8)^2;
32 end
33
34 parameters0 = [D0, S0]
35 [xMinSearch,fval,exitflag,output] =
36     fminsearch(@costFunction,parameters0);

```

```

30
31 Ef = xMinSearch(1)/(tf*d^2/2)
32 Gc = xMinSearch(2)/(d^2/tc)

```

An example of the collected data is shown:

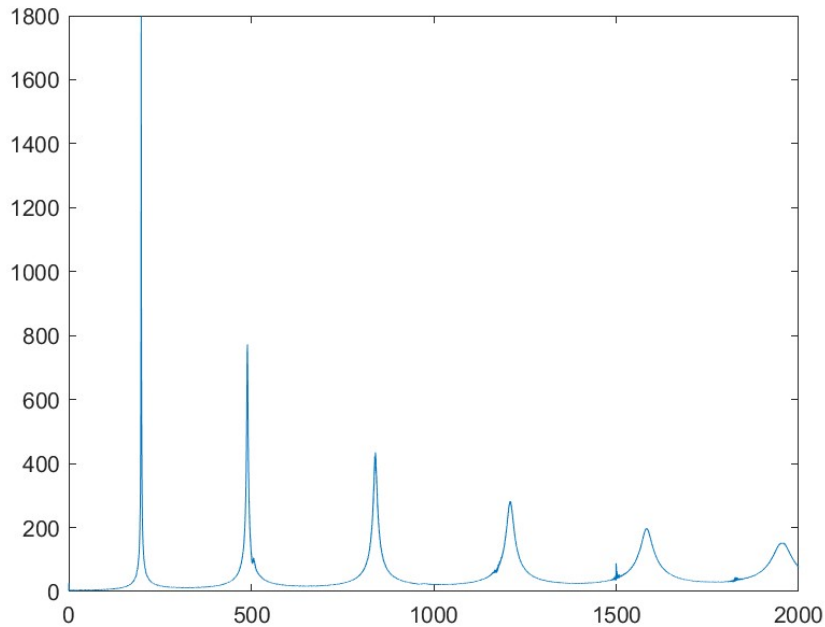


Figure 7: Experimental test results

where the frequency is represented on the x-axis and the frequency response function (FRF) on the y-axis.

The first two natural frequencies, corresponding to the first two peaks in Figure 7, are retrieved for each of the five tests undertaken. Eventually, a mean value for both is computed, in order to minimize the error that comes from the measurements:

$$\omega_1 = 198.7Hz, \quad \omega_2 = 489.8Hz$$

The natural frequencies of a beam are strictly dependent on its materials properties E_f, G_c , by means of the matrix A , as highlighted in Equation 10.

The idea is to find a combination of properties values, that return the same natural frequencies as the experimental ones, that represent the reference. For this sake, the following cost function is set up:

$$J(D, S) = (\omega_1 - \omega_1^e)^2 + (\omega_2 - \omega_2^e)^2 \quad (11)$$

And the minimization problem is solved:

$$[D^*, S^*] = \operatorname{argmin}[J(D, S)] \quad (12)$$

The adopted solver is the same used in Section 2.2: `fminsearch`. The starting values choice is simply based on similar material properties data searched from the web:

$$E_{F0} = 130 \cdot 10^9 \text{N/m}^2, \quad G_{C0} = 15 \cdot 10^6 \text{N/m}^2;$$

$$D_0 = E_{F0} \cdot t_F \cdot d^2/2 = 31930 \text{N} \cdot \text{m}, \quad S_0 = G_{C0} \cdot d^2/t_C = 402910 \text{N/m}$$

Since the cost function requires to find the natural frequencies for each combination of material properties, the Matlab integrated function `fzero`, applied on Equation 10, is exploited.

It requires an initial guess of frequencies: a good combination is given by the Euler frequency and its double: $\omega_1^E = 22.4 \cdot \sqrt{\frac{D}{\rho^* \cdot L^4}}$, $\omega_2^E = 2 \cdot \omega_1^E$.

Since data collected and validated from a previous test are available, it is even possible to verify the reliability of this algorithm: The retrieved results converge to the validated ones, confirming its effectiveness.

3.4 Results

The resulting values of bending and torsional stiffness (D, S) are retrieved, and the faces and core material properties (E_f, G_c) are consequently computed:

$$D = 11032 \text{N} \cdot \text{m}, \quad S = 1874219 \text{N/m};$$

$$E_F = \frac{D}{t_F \cdot d^2/2} = 44.9 \cdot 10^9 \text{N/m}^2, \quad G_C = \frac{S}{d^2/t_C} = 69.8 \cdot 10^6 \text{N/m}^2$$

3.5 Discussion

It's important to highlight that the face material is carbon fiber, that is usually strongly anisotropic, but in this case, the fibers are oriented in such a way to keep its properties quite independent from the considered direction.

In detail, the stress induced in the face, during the experiment, is almost considerable as tensile or compressive (depending on the considered beam side); thus the elastic modulus is defined in that direction.

4 Task 3 - Sandwich plate analysis

The goal of task 3 is to retrieve the sandwich panel response, in terms of deflection (w), normal stress in the face (σ_f) and shear stress in the core (τ_c), when subjected to a uniform pressure (P).

The panel is considered to have partially fixed edges: its boundary conditions will be useful later on.

The sandwich panel used for this task is the same as the one used later on in Section 6, being part of a ship hull.

4.1 Theory

During a preliminary design phase, it is reasonable to split the entire hull into smaller sub-panels to be analyzed.

Furthermore, the load resulting from the impact of the hull with the waves is considered as uniformly distributed on the surface.

4.2 Experimental setup

As study case for this task, the constitutive material properties are the ones retrieved in Section 3.4, while the load case and the panel dimensions are the ones considered later on in Section 6.

The resulting pressure is related to a design acceleration $a_{cg} = 30m/s^2$, and it is computed by means of this equation: $P = 80 \cdot A^{-0.3} [kN/m^2]$, (where $A = a \cdot b$ is the panel surface).

The panel analyzed in this section is represented here:

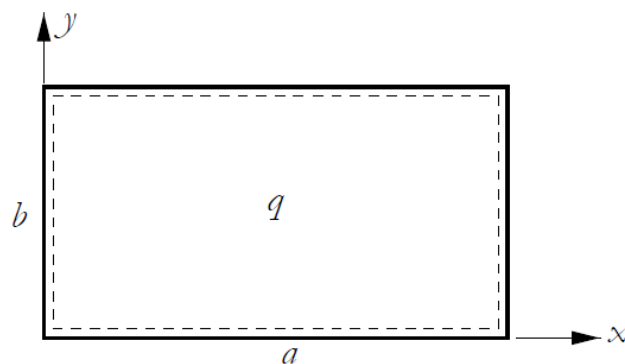


Figure 8: Panel representation

Where: $a = 1.75m$, $b = 2.25m$.

4.3 Method

The procedure to evaluate the panel response is entirely indicated by the DNV rules, thus it is only referenced in the current Section and not furthermore discussed. For the sake of this analysis, the involved rules are from B200 to B400.

The response values are evaluated for both the long and the short side of the panel, but only the maximum result for each response value is selected, as worst case.

A Matlab function used to evaluate the panel response values is developed and, to summarize, its inputs and outputs are listed here:

- **Inputs:** face thickness (t_f), core thickness (t_c), pressure acting on the panel (P), face elastic modulus (E_f), core shear modulus (G_c), panel short side (a), panel long side (b), Poisson ratio (ν);
- **Outputs:** panel deflection (w), normal stress in the face (σ_f), shear stress in the core (τ_c).

4.4 Results

To ease the analysis of the results, all the inputs for both the example and the chosen study case are collected in a table:

| Output | Example | Study case | Unit measure |
|--------|-----------------|--------------------|--------------|
| t_f | 4 | 0.72 | <i>mm</i> |
| t_c | 50 | 25.4 | <i>mm</i> |
| P | 220 | 53.03 | <i>kPa</i> |
| E_f | $70 \cdot 10^3$ | $44.92 \cdot 10^3$ | <i>MPa</i> |
| G_c | 80 | 69.78 | <i>MPa</i> |
| a | 1.8 | 1.75 | <i>m</i> |
| b | 1.2 | 2.25 | <i>m</i> |
| ν | 0.3 | 0.3 | — |

Table 6: Function inputs

The panel response values are collected in a table as well:

| Output | Example | Study case | Unit measure |
|---------------|----------------|-------------------|---------------------|
| w | 12.39 | 32.15 | mm |
| σ_f | 97.67 | 382.19 | MPa |
| τ_c | 2.06 | 1.42 | MPa |

Table 7: Panel response values

4.5 Discussion

It is visible how the study case panel presents much higher response values. This can be motivated by the choice of the materials, which are overall less rigid and also less thick in the sandwich build-up, in addition to a wider area.

5 Task 4 - Sandwich hull design

In the next tasks, part of the structural design of the Swedish Coast Guard vessel KBV 312 is undertaken.



Figure 9: KBV 312

In particular, a panel of the hull is designed. It is positioned between the coordinates $x = 6m$ and $x = 7.75m$, with respect to the following picture:

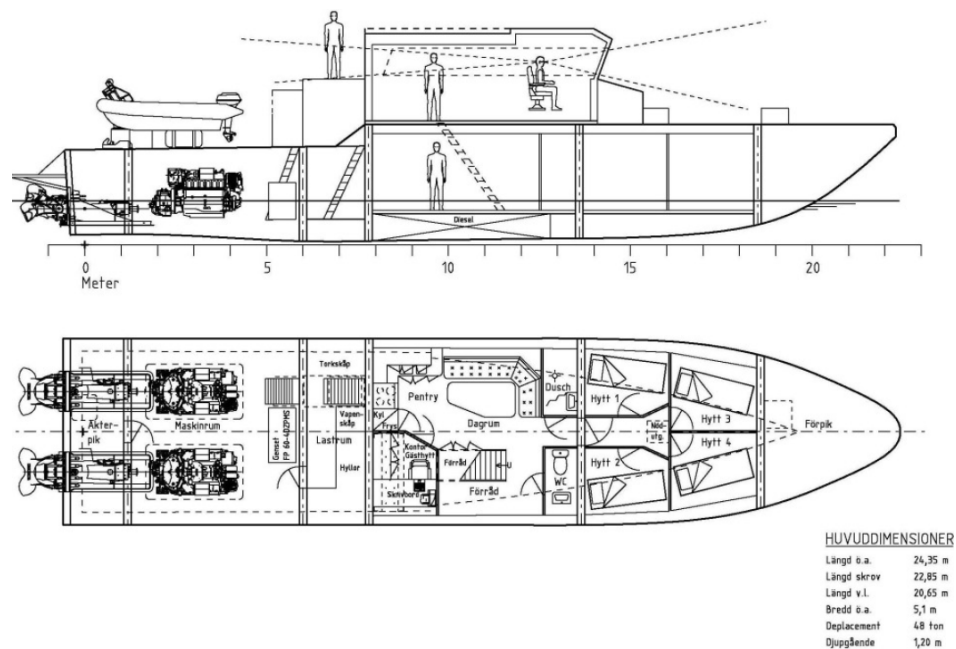


Figure 10: KBV 312 technical drawing

6 Task 4.1 - Simple panel optimization

The goal of this task is to optimize the weight of a panel making up the ship hull, ensuring that its response to the design load fulfills the set requirements. The design variables are the faces thickness (t_f) and the core thickness (t_c).

Firstly, the panel must fulfill only a maximum displacement requirement. Then, other requirements concerning the maximum normal stress at face level, the maximum shear stress at core level and the allowed buckling stress at face level must be fulfilled as well.

6.1 Theory

As explained in Section 4, a single representative panel is designed, and the pressure is considered to be uniformly distributed on its surface.

6.2 Experimental setup

As introduced in Section 4.2, the study case is a rectangular sandwich panel, characterized by the material properties retrieved in Section 3.4.

All its properties, along with the withstood pressure, are listed in the following table:

| Property | Value | Unit measure |
|----------|--------------------|--------------|
| a | 1.75 | m |
| b | 2.25 | m |
| E_f | $44.92 \cdot 10^9$ | N/m^2 |
| G_c | $69.78 \cdot 10^6$ | N/m^2 |
| E_c | $0.28 \cdot 10^9$ | N/m^2 |
| ν | 0.3 | – |
| ρ_f | 1500 | kg/m^3 |
| ρ_c | 155.59 | kg/m^3 |
| P | 53.03 | kPa |

Table 8: Sandwich properties

6.3 Method

To reach the goal of this task, to design the faces and core thicknesses, this procedure is followed:

1. An optimization problem is set up: $J(t_f, t_c) = t_c \cdot \rho_c + 2 \cdot t_f \cdot \rho_f$;
2. The constraint functions are defined:
 - (a) $g_1 = w_{Response} - w_{MAX}$, when only the displacement constraint is set;
 - (b) $g_1 = w_{Response} - w_{MAX}$, $g_2 = \sigma_{f,Response} - \sigma_{f,MAX}$, $g_3 = \sigma_{f,Response} - \sigma_{cr}$,
 $g_4 = \tau_{c,Response} - \tau_{c,MAX}$, when also the other constraints are set;
3. Minimum and maximum thicknesses values are established:

$$(t_f, t_c)_{MIN} = (0.5, 3)mm, \quad (t_f, t_c)_{MAX} = (2, 40)mm;$$
4. Initial guess of thicknesses for the Matlab algorithm: $(t_f, t_c)_0 = (1.5, 10)mm$;
5. The problem is solved by means of the Matlab built-in function `nlincon`.

The cost function variable $J(t_f, t_c)$ directly corresponds to the panel areal density.

The maximum values present in the constraint functions are computed as indicated in the project instructions according to the following formulas:

$$w_{MAX} = 0.02 \cdot b;$$

$$\sigma_{f,MAX} = \frac{E_f \cdot 10^{-6}}{77} \cdot FOS_f [MPa];$$

$$\tau_{c,MAX} = \left(\frac{\rho_c^2}{60788} + \frac{\rho_c}{70.6} - 0.1912 \right) \cdot FOS_c [MPa]$$

For what concerns the wrinkling load, instead it is a well-known equation from theory:

$$\sigma_{cr} = 0.5 \cdot (E_f \cdot E_c \cdot G_c \cdot 10^{-18})^{1/3} \cdot FOS_f [MPa]$$

where $FOS_f = \frac{1}{0.3}$, $FOS_c = \frac{1}{0.4}$ are safety factor values suggested in the project instructions. These coefficients are introduced to tackle uncertainties, manufacturing errors and other approximations that would lead to an under-estimation of the thicknesses.

For what concerns the response values, the procedure to compute them has already been explained in Section 4.3, where a function undertaking this job has been introduced. Now, it has been enough to expand that function, introducing the equations at point 2.

The minimum thicknesses values are chosen to avoid the design colliding with manufacturing issues related to too small laminates or cores. The maximum values, instead, are chosen to avoid having excessively bulky structures.

The initial guess values are simply chosen on the basis of reasonable evaluations.

6.4 Results

In Table 9 are reported the design variables for the following two cases:

- Only the displacement constraint is set;
- The extended constraint set is considered.

| Thickness | Displacement | Extended set | Unit measure |
|-----------|--------------|--------------|--------------|
| Face | 0.65 | 0.65 | <i>mm</i> |
| Core | 25.52 | 25.52 | <i>mm</i> |

Table 9: Design variables

Other relevant results coming from the optimization problems resolution are:

- Cost function value at the initial guess (sandwich areal density);
- Cost function value at the optimal point;
- Number of iterations to reach the optimal point;
- Constraint function values at the optimal point;

They are collected in a table for the two different constraint cases:

| Result | Displacement | Extended set | Unit measure |
|-----------------------|-----------------------|--------------------|-------------------------|
| Initial areal density | 6.06 | 6.06 | <i>kg/m²</i> |
| Optimal areal density | 5.91 | 5.91 | <i>kg/m²</i> |
| Number of iterations | 16 | 272 | – |
| g_1 | $-2.51 \cdot 10^{-8}$ | ~ 0 | <i>m</i> |
| g_2 | – | $-1.52 \cdot 10^3$ | <i>MPa</i> |
| g_3 | – | $-1.17 \cdot 10^3$ | <i>MPa</i> |
| g_4 | – | $-0.01 \cdot 10^3$ | <i>MPa</i> |

Table 10: Optimization results

It is interesting to graphically represent the optimization problem. To do so, the trend of the cost function and of the constraint functions with respect to the thicknesses is visualized, and the design point updating during the optimization is kept track of.

As done before, two plots are shown for both the constraint cases. The simple displacement case is represented here:

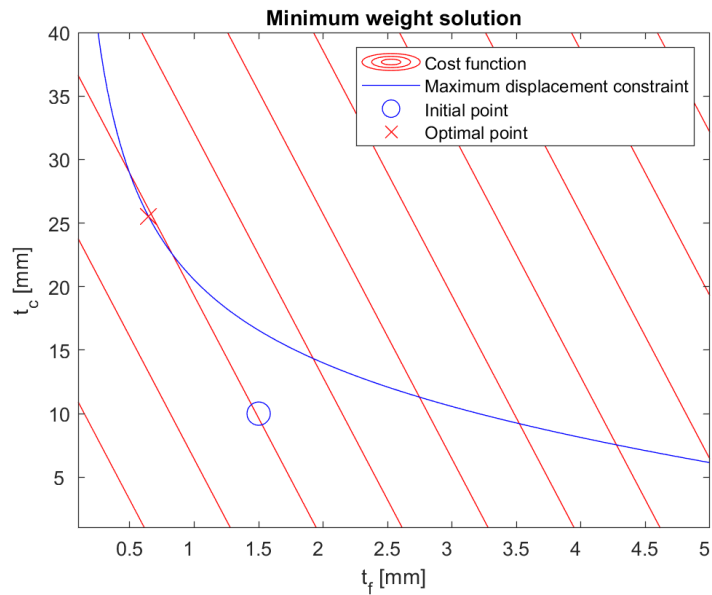


Figure 11: Functions representation for the basic constraint set

The extended constraint set case is shown here:

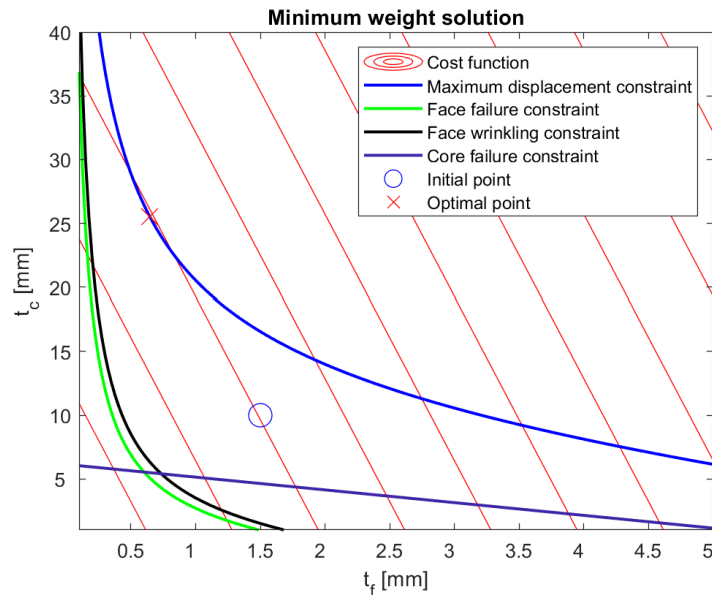


Figure 12: Functions representation for the extended constraint set

6.5 Discussion

The most visible outcome from the results is that both the two cases lead to the same thicknesses values, and so to the same optimality conditions.

The reason why this happens is evident when looking at the constraint functions: both constraint functions, relative to the maximum allowable displacement, present very low values close to 0. This means that they are active in the solution: they limit the optimization problem by finding a further optimal solution.

The other three functions in the extended constraint set case present much more negative values, meaning that the response values satisfy the maximum conditions but are not close to them. Thus, they are not active in the solution.

An evidence to all these observations is given by the plots of Figure 11 and 12, where the optimal point lays on the maximum displacement constraint function.

Considering the lower and upper thickness limits established in Section 6.3, the optimal thicknesses don't reach these limits, meaning that neither them are active.

The number of iterations to reach the optimality is very low for the first case (16) where only one constraint is set, while it significantly increases to 272, when the number of constraint functions increases as well. This trend can be motivated by the optimality search algorithm, that gets more computationally demanding when it has to satisfy more constraints at the same time.

7 Task 4.2 - Panel with stiffener optimization (basic)

The goal of this task is to design a stiffener to implement in the panel to increase its strength. Some structural constraints must be verified, and at the same time, the overall panel weight must be optimized.

7.1 Theory

The theory is based on the same assumptions introduced in Section 4.1: analysis undertaken on a panel of the hull and uniform pressure across the panel.

The response of the I-beam shaped stiffener isn't anymore described by the DNV rules as for the panel, but it is well defined from theory. The idea is to move from a more complex I-beam to an equivalent simple beam characterized by the same bending and shear stiffness. Considering a partially fixed beam as the one depicted here:

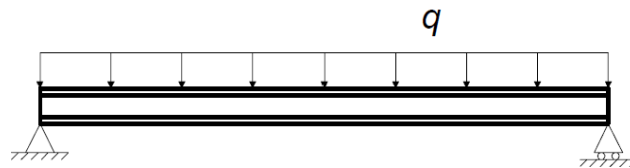


Figure 13: Partially fixed beam

the displacement is known and can be retrieved from the following equation:

$$w(x) = w_b(x) + w_s(x) = \frac{q \cdot a^4}{24D} \left[\left(\frac{x}{a}\right)^4 - 2 \left(\frac{x}{a}\right)^3 + \left(\frac{x}{a}\right) \right] + \frac{q}{2S}(a \cdot x - x^2) \quad (13)$$

where the stiffness contributions are introduced later in Section 7.3. Analyzing this equation, it is evident how the displacement is separately made up of the bending and of the shear contributions. The maximum deflection is expected to be at half of the beam, thus the x coordinate used inside Equation 13 corresponds to half the original panel shortest side ($a/2$). It is assumed that the reaction to the load can be studied separately for the sandwich panel and the stiffener. Thus, the stiffness and the strength constraints are separately verified. Since the stiffener is joined to the face panel, the stress deriving from the load it withstands is partially distributed on the upper face panel. The region of influence is identified by the parameter b_{ef} (effective breadth of flange):

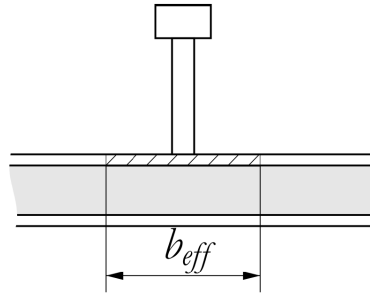


Figure 14: Stiffener front view

The computation of this parameter is explained in the next Section 7.2.

7.2 Experimental setup

The web stiffener is introduced at the half of the panel, as represented in the following picture:

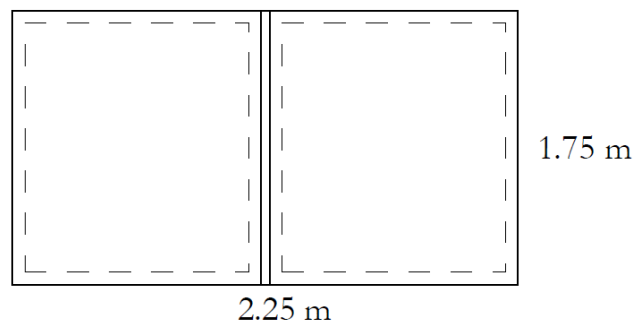


Figure 15: Stiffener implementation

Its characteristic dimensions, to be optimized, are reported in the following picture:

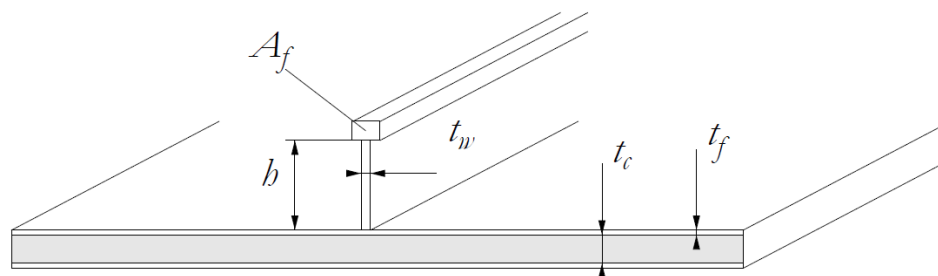


Figure 16: Stiffener representation

The effective breadth of flange b_{eff} is calculated by the following equation:

$$b_{eff} = \frac{b}{1 + 10 \cdot (b/l)^2} \quad (14)$$

where:

- $b = 2.25m$: panel breadth between beams;
- $l = 0.58 \cdot a$: stiffener effective length. The real length is multiplied by a factor, due to the beam boundary condition of fixed ends.
- $a = 1.75m$: stiffener length.

Now that all the study-case parameters are introduced, it is possible to calculate the bending and shear stiffnesses:

$$D = E_f \cdot \frac{A_f \cdot b_{eff} \cdot t_f \cdot h^2}{A_f + b_{eff} \cdot t_f}; \quad S = G \cdot t_w \cdot h$$

7.3 Method

The panel is split into two halves to satisfy the decoupling condition related to the stiffness constraints for the beam and the panel. Take into account that the optimization process is jointly undertaken.

In this study-case, the panel area, where the load is distributed, is halved with respect to the one in Section 6.2:

$$P = 80 \cdot \left(a \cdot \frac{b}{2}\right)^{-0.3}$$

Whereas it keeps the same for the beam load.

The stiffness constraint for the stiffener is introduced by means of a maximum displacement, as previously done:

$$w_{s,MAX} = 0.005 \cdot b$$

The response of the beam (Section 6.3) is compared to this maximum deflection in one of the two constraint functions involved in this optimization problem:

$$g_1 = w_s - w_{s,MAX} \tag{15}$$

The other stiffness constraint function concerns the panel, as explained in Section 6.3:

$$w_{p,MAX} = 0.02 \cdot b$$

$$g_2 = w_p - w_{p,MAX} \tag{16}$$

Finally, it is necessary to choose lower and upper limits for the optimization problem variables, and initial guess values. The choice of these parameters is mainly based on manufacturability

of the sandwich panel and on the higher cost related to the consumption of carbon fiber material.

| Parameter | Lower limit | Upper limit | Initial guess | Unit measure |
|-----------|-------------|-------------|---------------|--------------|
| t_f | 0.5 | 2 | 1.5 | mm |
| t_c | 5 | 40 | 10 | mm |
| A_f | 10 | 100 | 50 | mm^2 |
| h | 2 | 50 | 10 | mm |
| t_w | 0.5 | 5 | 3 | mm |

Table 11: Optimization limits and initial guess

7.4 Results

The results from the optimization problem solving are reported in the following table:

| Parameter | Optimized value | Unit measure |
|-----------|-----------------|--------------|
| t_f | 0.75 | mm |
| t_c | 28.61 | mm |
| A_f | 10.1 | mm^2 |
| h | 2.11 | mm |
| t_w | 0.53 | mm |

Table 12: Optimization results

The resulting cost function value is the areal density of the panel with the stiffener:

$$\rho^* = \rho_f \cdot t_f + \rho_c \cdot t_c + \rho_f \cdot \frac{A_f + t_w \cdot h}{b} = 6.71 kg/m^2 \quad (17)$$

Its value at the initial guess is: $\rho_0^* = 6.11 kg/m^2$.

The constraint function values at the initial guess and at the optimality point are:

| Constraint function | Initial guess | Optimal point | Unit measure |
|---------------------|---------------|---------------|--------------|
| Panel displacement | 76.1 | ~ 0 | mm |
| Beam displacement | -8.8 | -8.6 | mm |

Table 13: Constraint function values

7.5 Discussion

It is noticeable how the cost function value at the optimal point is lower than the value at the initial guess. This is related to the fact that the initial guess point represents an unfeasible solution for the optimization problem, due to the constraint functions. Indeed, one of them is positive, while both of them are negative at the optimal point, standing for a final feasible solution. In more detail, the constraint function for the panel is approximately 0, meaning that this constraint is active for the solution.

8 Task 4.3 - Panel with stiffener optimization (extended)

The goal of this task is to extend the optimization problem undertaken in the previous Section 7, by implementing the constraints relative to the strength of the panel and the stiffener.

A further constraint is lately introduced to account for the minimum quantity of reinforcement for the carbon fiber face panel.

8.1 Theory

The stiffener is treated as a partially fixed beam, as previously done in Section 7.1, where the load is uniformly distributed on its surface.

The I-beam equivalence of the stiffener behaves similarly to a sandwich panel when dealing with stress states. Indeed, it is supposed that the two flanges only carry normal stress, while the web carries shear stress.

The maximum bending moment and the shear force generating the highest stress state inside the stiffener are:

$$M_{s,MAX} = q \cdot \frac{a^2}{4} - q \cdot \frac{a^2}{8}; \quad T_{s,MAX} = q \cdot \frac{a}{2} \quad (18)$$

Considering a sandwich beam stress response, the stress state inside the stiffener is thus described in the following way:

$$\begin{aligned} \sigma_s &= \frac{M_{s,MAX}}{D} \cdot E_f \cdot \frac{h}{2}; \\ \tau_s &= \frac{T_{s,MAX}}{D} \cdot E_f \cdot \frac{t_f \cdot h}{2} \end{aligned} \quad (19)$$

where the stiffnesses are calculated in the same way as done in Section 7.3.

8.2 Method

New constraint functions are introduced with respect to the previous Section 7.3, but the same optimization variable limits and initial guess are used.

The extended set of constraint functions now becomes:

$$\begin{aligned}
g_1 &= w_p - w_{p,MAX}; \\
g_2 &= w_s - w_{s,MAX}; \\
g_3 &= \sigma_p - \sigma_{f,MAX}; \\
g_4 &= \sigma_s - \sigma_{f,MAX}; \\
g_5 &= \sigma_p - \sigma_{cr}; \\
g_6 &= \sigma_s - \sigma_{cr}; \\
g_7 &= \tau_p - \tau_{c,MAX}; \\
g_8 &= \tau_s - \tau_{f,MAX}
\end{aligned} \tag{20}$$

where:

- the maximum deflections $w_{p,MAX}$ and $w_{s,MAX}$ are those introduced in the previous Section 7.3;
- the face tensile strength $\sigma_{f,MAX}$ and wrinkling load σ_{cr} , and the core shear strength calculations are showed in Section 6.3;
- the web shear strength is calculated as: $\tau_{c,MAX} = 0.05 \cdot \sigma_{f,MAX}$.

After a first optimization problem solving with this constraints set, a further constraint relative to the minimum amount of reinforcement is introduced. According to the DNV rules, the minimum areal density for a carbon fiber sandwich panel face outside of the hull is:

$$\rho_{r,MIN}^* = \rho_{r,0}^* \cdot (1 + k \cdot (L - 20)) \tag{21}$$

where $L = 23m$ is the ship length, and $\rho_{r,0}^* = 1600g/m^2$ and $k = 0.025$ are two tabled data from DNVs.

It is assumed that the amount of reinforcement in the composite material is approximately the 55% in weight. Starting from this information, it is possible to come up with the minimum face thickness in the sandwich panel:

$$M_r = 0.55 \cdot M_f$$

where M_r is the mass of reinforcement and M_f is the face mass;

$$A_p \cdot \rho_{r,MIN}^* = 0.55 \cdot \rho_f \cdot A_p \cdot t_{f,MIN}$$

where A_p is the surface of the panel, ρ_f is the face volumetric density and $t_{f,MIN}$ is the minimum thickness of the face;

$$t_{f,MIN} = \frac{\rho_{r,MIN}^*}{0.55 \cdot \rho_f} = 2.08mm$$

Now it is possible to update the previous requirement for the minimum face thickness.

8.3 Results

The results of the optimization problems, firstly imposing only the strength constraints and then adding the reinforcement constraint, are collected in the following table:

| Parameter | Starting set | Expanded set | Unit measure |
|------------|--------------|--------------|--------------|
| ρ^* | 6.71 | 8.80 | kg/m^2 |
| ρ_0^* | 6.11 | 6.11 | kg/m^2 |
| t_f | 0.75 | 2.08 | mm |
| t_c | 28.61 | 16.31 | mm |
| A_f | 10 | 5 | mm^2 |
| h | 2 | 2 | mm |
| t_w | 0.5 | 0.5 | mm |

Table 14: Optimization results

In the same way, the constraint function values at the optimal point are shown:

| Parameter | Starting set | Expanded set | Unit measure |
|-----------|--------------|--------------|--------------|
| g_1 | ~ 0 | ~ 0 | mm |
| g_2 | -8.54 | -8.75 | mm |
| g_3 | -1.69 | -1.80 | GPa |
| g_4 | -1.94 | -1.94 | GPa |
| g_5 | -1.28 | -1.39 | GPa |
| g_6 | -1.53 | -1.53 | GPa |
| g_7 | -5.0 | -4.3 | MPa |
| g_8 | -97.2 | -97.2 | MPa |

Table 15: Constraint function values

8.4 Discussion

Beginning from a comparison of the results in this Section with the ones in the previous Section 7.4, the cost function value at the optimal point is always the same ($\rho^* = 6.71\text{kg}/\text{m}^2$) when only implementing the strength constraint functions. Indeed, the only active constraint is the one concerning the stiffness, as in the previous case.

Instead, when implementing the constraint relative to the minimum amount of reinforcement, the cost function value increases by the 31.1%. This is justified by the reinforcement constraint itself, that becomes active together with the stiffness one.

The visible result is that the face thickness increases to satisfy the constraint on its minimum value, moving from 0.75mm to 2.08mm . Thanks to the optimization problem, it is possible to recover this increase of dimension and so of weight. Indeed, the core thickness decreases from 28.61mm to 16.31mm .

9 Task 4.4 - Sandwich hull design

The goal of this task is to find the optimal number of stiffeners in the panel, ensuring that all the constraints introduced until now are still verified.

9.1 Experimental setup

The setup is always the same as in the previous tasks, but at this time the number of stiffeners can vary:

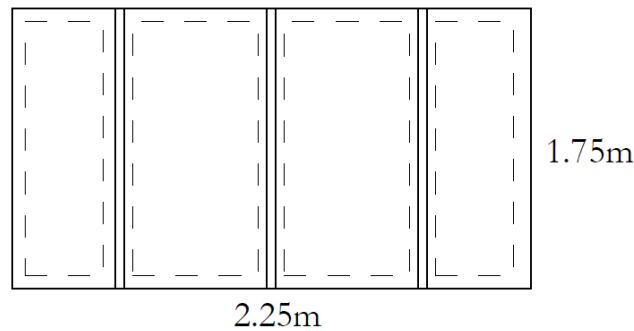


Figure 17: Panel with different stiffeners

9.2 Method

The objective function is now dependent on the number of stiffeners:

$$\rho^* = \rho_f \cdot t_f + \rho_c \cdot t_c + n \cdot \rho_f \cdot \frac{A_f + t_w \cdot h}{b} \quad (22)$$

where n is the number of stiffeners.

When increasing this number, the load that the panel and the single stiffener itself carry diminishes accordingly:

$$\begin{aligned} P_{panel} &= 80 \cdot \left(a \cdot \frac{b}{2 \cdot n} \right)^{-0.3} \\ P_{stiffener} &= 80 \cdot \left(a \cdot \frac{b}{n} \right)^{-0.3} \end{aligned} \quad (23)$$

where a and b are respectively the shortest and the longest panel dimensions.

The optimization process is undertaken 10 times, considering an increasing number of stiffeners. Finally, the number corresponding to the lowest overall areal density is chosen as the design value.

9.3 Results

For each number, the optimization results are shown:

| Parameter | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Unit measure |
|-----------|-------|------|------|------|------|------|------|------|------|------|--------------|
| ρ^* | 8.80 | 6.79 | 6.58 | 6.58 | 6.59 | 6.59 | 6.59 | 6.60 | 6.60 | 6.60 | kg/m^2 |
| t_f | 2.08 | 2.08 | 2.08 | 2.08 | 2.08 | 2.09 | 2.08 | 2.08 | 2.08 | 2.08 | mm |
| t_c | 16.31 | 3.39 | 2.00 | 2.00 | 2.00 | 2.01 | 2.00 | 2.00 | 2.00 | 2.00 | mm |
| A_f | 5.00 | 5.15 | 5.10 | 5.00 | 5.00 | 5.20 | 5.01 | 5.01 | 5.00 | 5.01 | mm^2 |
| h | 2.00 | 2.27 | 2.18 | 2.00 | 2.00 | 2.36 | 2.02 | 2.02 | 2.00 | 2.01 | mm |
| t_w | 0.50 | 0.57 | 0.55 | 0.50 | 0.50 | 0.58 | 0.50 | 0.50 | 0.50 | 0.50 | mm |

Table 16: Optimization results

The constraint conditions are verified for each undertaken simulation.

9.4 Discussion

Increasing the number of stiffeners, the load that the panel and the stiffeners themselves carry decreases. Thus, it is expected that, for a higher number, the optimization variables diminish until they get constrained by the chosen lower bounds.

This observation is validated by the results, that show a convergence to each parameter lower bound. This trend doesn't necessarily mean that the lower constraints are too strict. Instead, it can be a sign that increasing too much the number of stiffeners is not helpful, also because manufacturing issues may occur from the design of excessively small face sheets and core.

In the end, the number of stiffeners leading to the smallest panel area density is $n = 3$.