Romain PINQUIE

MSc Computational and Software Techniques in Engineering
Option: Computer Aided Engineering

Computational Engineering Design Optimisation

Assignment Report:

Optimisation of the amount of material used to build a column.

Academic Year 2011/2012

AMAC
School of Engineering
Cranfield University
Bedford
BK 43 0AL
United-Kingdom

Romain PINQUIE

MSc Computational and Software Techniques in Engineering
Option: Computer Aided Engineering

Romain PINQUIE

MSc Computational and Software Techniques in Engineering
Option: Computer Aided Engineering

SUMMARY

I. Overview

"As the world becomes economically richer, it becomes environmentally poorer". *Dr Norman Myers, Green College, Oxford*

We are leaving in a world where growth population, comfort improvements and renovation of old buildings force the companies of civil engineering to start thousands of new construction projects every day. In addition, the construction of a new building requires a lot of money and a huge amount of materials. Nevertheless, mineral and natural resources used to make materials are limited and people are hopefully recently environmentally aware and then try to reduce their fossil energy-consumption in order to save money and to protect the environment.

This dilemma is not only true for the population but also for companies, especially building companies. *Engineering Design Optimisation* of a building project can definitely enable companies to economize on material purchases and improve their environment impact. Therefore I have decided to work on the *optimisation of the amount of material used to build a column of a building*.

II. Problem

Stress and strength theory shows that a beam can carry several behaviours like bending, twisting, buckling, etc. However, here the optimisation has been done for a column subjects to an external load, thus this case study corresponds to a *buckling analysis* of a beam with a *fixed end*, i.e. fully fixed in the ground (cf. Figure 1).

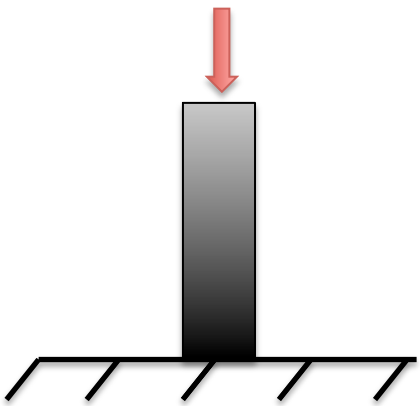


Figure 1 - Column and load

Civil engineers use to work with different kind of cross-sections (rectangular, cylindrical, H, etc.). The main goal is to reduce the amount of material used, then it seems obvious to carry out the research using a very *thin annulus column (cf. Figure 2),* which does not require not all that much material.

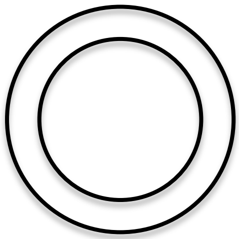


Figure 2: Thin annulus cross-section

III. Artefact model

1. Constants

Constants are the properties that do not change or influence the optimisation algorithm.

- Material density [ρ]

2. Parameters

Parameters are data susceptible to have a direct impact on the optimisation but it is not the case since they are fixed at the design stage, in most cases they are a constraints of the customers.

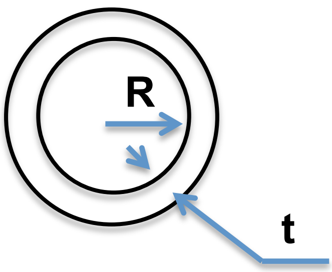
- Height [h]
- Load [P]
- Young's Modulus [E]

3. Design variables

The design variables correspond to the physical properties which impact directly on the amount of material used.

- Radius of small circle [R]

- Thickness of the annulus [t]



There are two continuous decisions variables, thus it is a multivariate problem. These variables are called independent because they do not have any relationship.

IV. Decision model

Computational Engineering Design Optimisation is based on the classic mathematical problem corresponding to the search of either a maximum or a minimum (according to the need) of a function called *"Objective function"*. Objective function is nothing less than the relationship between the design variables defined previously.

The purpose is to reduce the amount of material used to build a column. However, the height of the element is according to the customer's need(s) (architecture), thus in other words the main goal of the problem is to optimise (reduce) the cross-section area of the column. Therefore the minimum of the following equation must be found:

Single objective function: $\boldsymbol{A(t,R) = \pi[(R+t)^2 - R^2]}$

V. Constraints

Design constraints are defined by the "real-world", e.g. manufacturing processes force analysts to take in account the limit(s) of the machines or any physical dimension cannot be negative, etc. Design constraints should be obvious… Furthermore, design constraints also correspond to the limit(s) of the product/system itself; they are its limit of "good" working.

Mechanical design constraints:

- (c1) $R \geq 10$ cm: The radius must be up to 10 cm since a beam is seldom less than 10 cm in the real-word. Moreover, for this case study we will consider classic cross-sections, i.e. R = 10, 15, 20, 25, 30, …, 100 cm.

- (c2) $R \leq 100$ cm: Beam element's radius used to not be bigger than 1m.

- (c3) $t > 2.5$ cm: A limit of 2.5 cm of thickness has been decided than a smaller one might be a "silly" constraint. To match with the evolution of the radius the following thicknesses has been considered: t = 2.5, 5, 7.5, 10, 12.5, …, 25 cm.

- (c4) $R > t$: Because it has been assumed that the optimisation has been designed for thin annulus section.

Structural analysis constraints:

- (c5) $\sigma \leq \sigma_r$: Criteria of rupture in compression
- (c6) $\sigma \leq \sigma_{bg}$: Criteria of general buckling (elastic instability)
- (c7) $\sigma \leq \sigma_{bl}$: Criteria of local buckling

⇒ Rewriting equations (4) (5) (6) and (7) :

- (c4) $\frac{t}{R} - 1 \leq 0$
- (c5) $\frac{\sigma}{\sigma_r} - 1 \leq 0$
- (c6) $\frac{\sigma}{\sigma_{bg}} - 1 \leq 0$
- (c7) $\frac{\sigma}{\sigma_{bl}} - 1 \leq 0$

VI. Behaviour of the function

It is possible to perform some hand calculations in order to find the critical point(s) of the function and to get some informations about its position (maximum, minimum, saddle point).

Starting with: $\mathbf{W(t, R) = \pi[(R + t)^2 - R^2] = \pi t^2 + 2\pi} \boldsymbol{Rt}$

This function is everywhere differentiable, so extrema can only occur at point (t*,R*) such that **∇(t*,R*)=(0,0).**

Firstly it has been necessary to seek the coordinates of the critical point(s) using the gradient operator:

$$\boldsymbol{\nabla W(R, t)} = \begin{bmatrix} \dfrac{\boldsymbol{\partial W(t, R)}}{\boldsymbol{\partial t}} \\ \dfrac{\boldsymbol{\partial W(t, R}}{\boldsymbol{\partial R}} \end{bmatrix} = \begin{bmatrix} \mathbf{\pi(2t + 2R)} \\ \mathbf{2\pi t} \end{bmatrix}$$

Using **∇(t*,R*)=(0,0):**

$$\begin{cases} \pi(2t + 2R) = 0 \\ 2\pi t = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} 2\pi R = 0 \\ t = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{R = 0} \\ \mathbf{t = 0} \end{cases}$$

Hence the stationary point (t,R*) is (0,0), however since the boundary conditions require both R and t up to 0 cm, hence it does not matter here. Nevertheless, the Hessian analysis has been carried out by "curiosity".*

To get more information about the stationary point, Hessian matrix has been calculated:

$$\mathbf{H(t,R)} = \begin{bmatrix} \dfrac{\partial^2(t,R)}{\partial^2 t} & \dfrac{\partial^2(t,R)}{\partial t \partial R} \\ \dfrac{\partial^2(t,R)}{\partial R \partial t} & \dfrac{\partial^2(t,R)}{\partial^2 R} \end{bmatrix} = \begin{bmatrix} \mathbf{2\pi} & \mathbf{2\pi} \\ \mathbf{2\pi} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{H(t^*,R^*) = H(0,0) =} \begin{bmatrix} \mathbf{2\pi} & \mathbf{2\pi} \\ \mathbf{2\pi} & \mathbf{0} \end{bmatrix}$$

<u>*1st method:*</u>

Let H1 denote the first principal minor of H(0,0) and let H2 denote its second principal minor. Then, det (H1) = 2π and det (H2) = $2\pi * 0 - 4\pi^2 = -4\pi^2$. Thus det(H1) > 0 and det(H2) < 0, therefore **H(0,0) is indefinite** (neither positive definite nor negative definite). Hence the point **(0,0) is a saddle point.**

<u>*2nd method:*</u>

Then the nature of the Hessian matrix has been analysed by using its eigenvalues:

$$\boldsymbol{det\ (H)} = |\mathbf{A} - \boldsymbol{\lambda}.\mathbf{I}| = \begin{vmatrix} \mathbf{2\pi - \lambda} & \mathbf{2\pi} \\ \mathbf{2\pi} & \mathbf{-\lambda} \end{vmatrix} = \boldsymbol{\lambda^2 - 2\pi\lambda - 4\pi^2}$$

$$\Delta = 4\pi^2 + 16\pi^2 = 20\pi^2 > 0$$

Eigenvalues:

$$\boldsymbol{\lambda 1} = \frac{\mathbf{2\pi + \sqrt{20\pi^2}}}{\mathbf{2}} \cong \mathbf{10.16}$$

$$\boldsymbol{\lambda 1} = \frac{\mathbf{2\pi - \sqrt{20\pi^2}}}{\mathbf{2}} \cong \mathbf{-3.88}$$

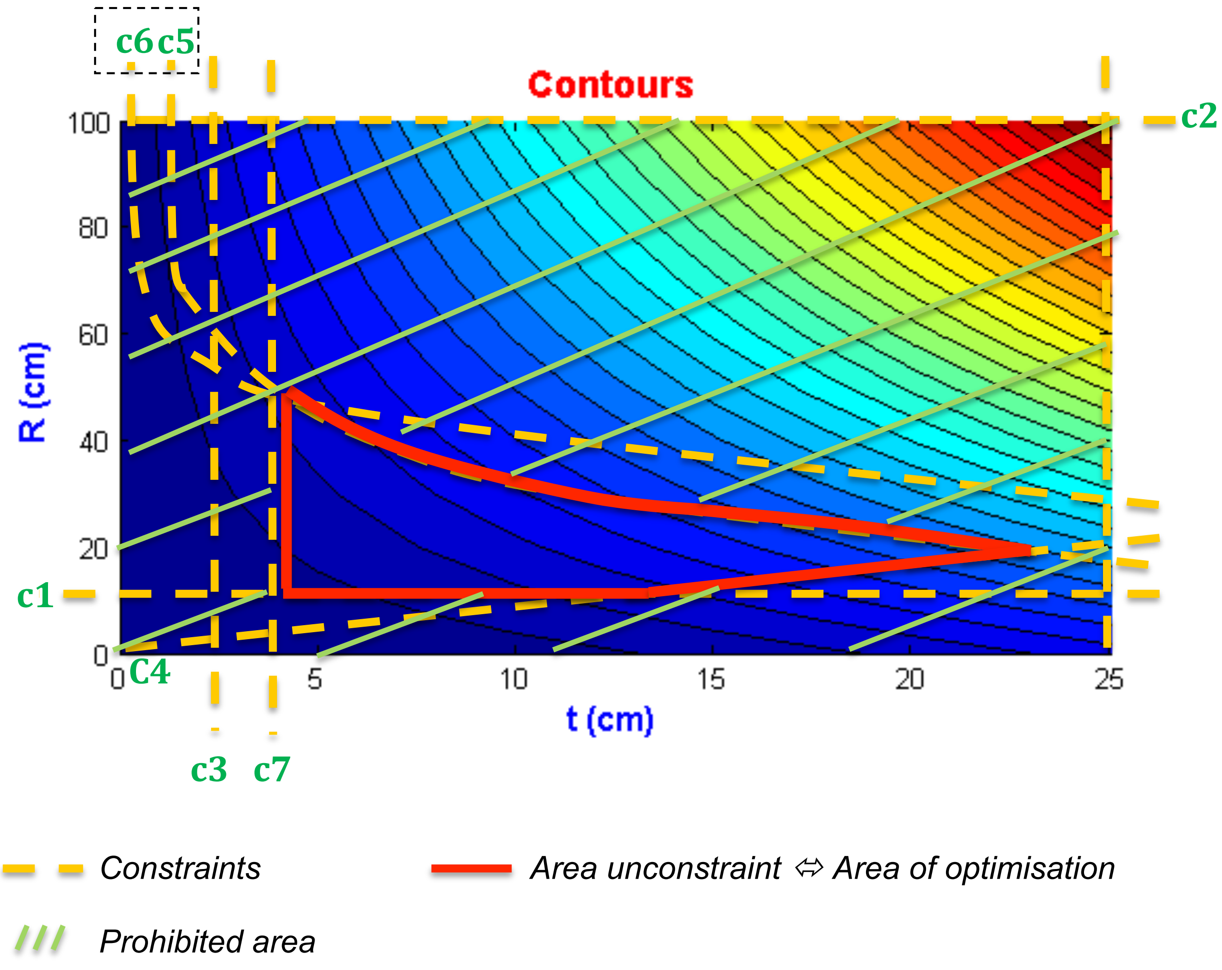
The eigenvalues are alternated, one is negative the other is positive, so H(0,0) is indefinite and the stationary point is a saddle point.

VII. Multi-dimensional problem

The objective function is a 2D function (t,R) which require a multi-dimensional search method in order to find the minimum of A(t,R). Before to look for the best appropriate algorithm to solve the problem it is essential to understand how the constrained problem works using the boundaries.

<u>*Contours:*</u>

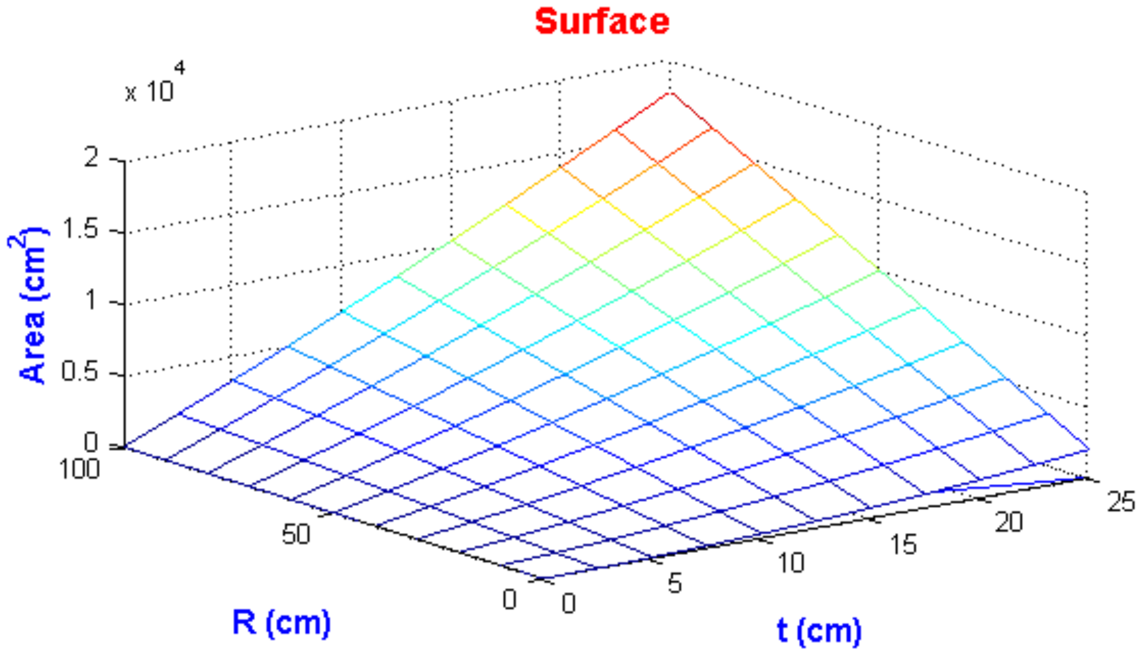
According to the constraints defined previously we can draw all of them and see which area remains available *(the quality of the constraints is not relevant and it does not matter as the main goal of this assignment is to show a good understanding of optimisation, further information in Appendix 1).*



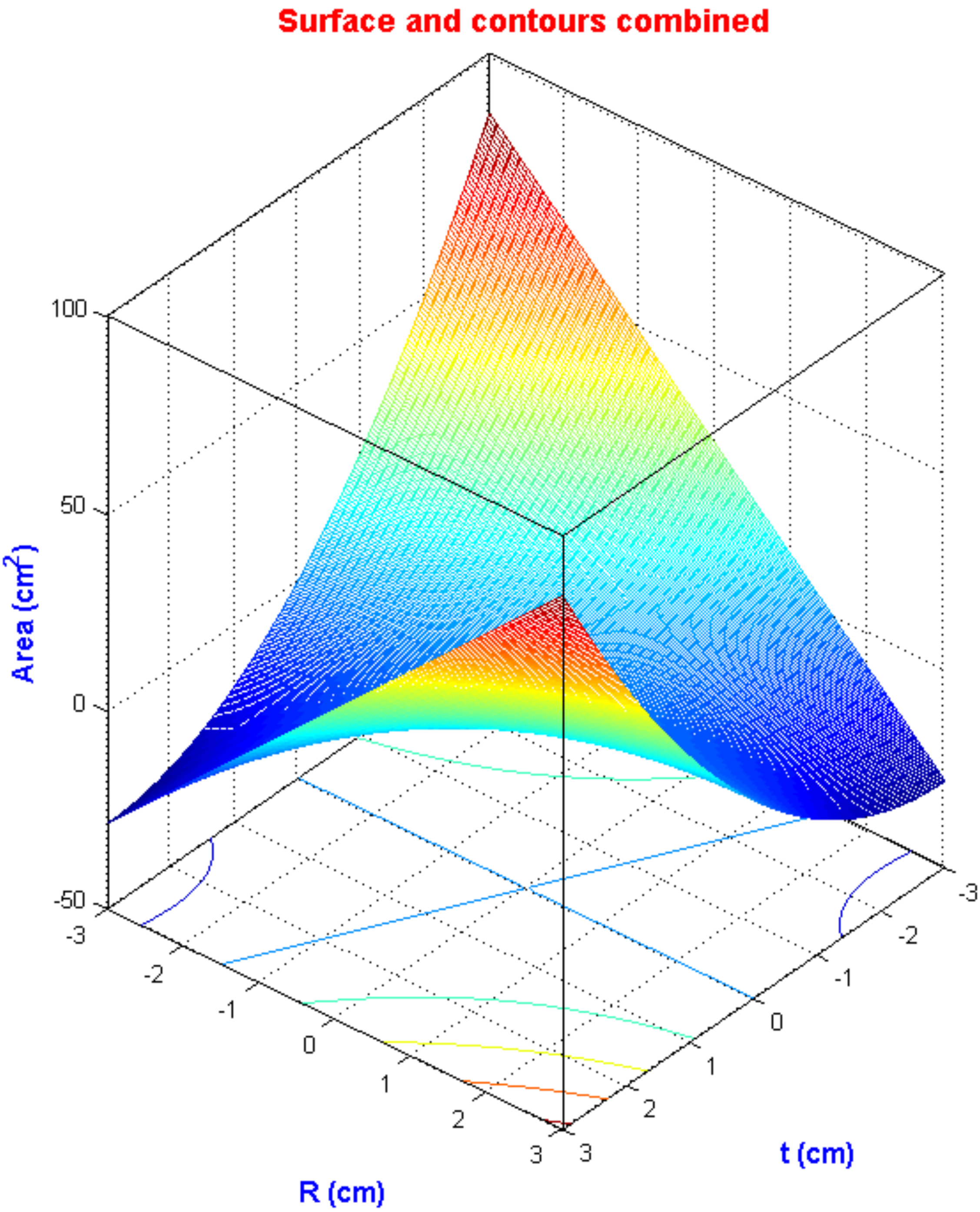
The contours of the function A(t,R) enable to see the gradients and the isolines of the quadratic function. Then using the constraints defined before the main area has been trimmed and at the end only one part remains available. We will see further away how implement a search method which allows seeking extremum.

Surface:

The surface below shows the evolution over $t \in [0;25]$ and $R \in [0;100]$ of the cross-section's area (relative to the weight) in function to both variables the thickness (t) and small radius (R).



<u>*Contours and surface combined:*</u>



The figure above represents the objective function A(t,R) in three dimensions and its contours on a bottom plane. The hand calculations performed at the previous chapter are confirmed since it is obvious to notice a saddle point at (0,0,0). Please be aware of the boundary conditions, i.e. that the surface makes sense only for positive value of t and R, but they have been extended to negative values in order to see the saddle point easily.

VIII. Optimisation algorithm

The problem includes two independent variables, for this reason a Multi-dimensional search method had to be chosen. There are three main Multi-dimensional search methods: *Random, Direct Search and Gradient Search.* The gradient of the objective function was available, thus Gradient Search method was the most efficient to solve the problem. In addition, Gradient search methods include: *Steepest Descent, Newton, Newton-Raphson and Conjugate Gradient Methods.* The method of steepest descent does not require Hessian matrix but its rate of convergence is slow (linear). Newton's method offers a quicker convergence (quadratic) than steepest descent since it uses gradient information and the second-order information. Newton-Raphon's method offers a quadratic convergence as well, but possible indefinite oscillation is avoided. The last Multi-dimensional search method corresponds to the Conjugate Gradient which is an improved steepest descent method offering faster convergence and less CPU time as there is no need to invert Hessian matrix. However the objective of this problem is quadratic, thus convergence can be reached in one step using Newton's method and in two steps with Conjugate Gradient method (number of iteration equals to number of variables, i.e. two in this case). Therefore Newton's method is more efficient here. Let see how it works:

<u>*Objective function:*</u> $\mathbf{W(t, R) = \pi t^2 + 2\pi} \boldsymbol{Rt}$

<u>*Newton's Method:*</u>

$$\boldsymbol{\nabla W(R, t)} = \begin{bmatrix} \mathbf{\pi(2t + 2R)} \\ \mathbf{2\pi t} \end{bmatrix}$$

$$\mathbf{H(t, R)} = \begin{bmatrix} \mathbf{2\pi} & \mathbf{2\pi} \\ \mathbf{2\pi} & \mathbf{0} \end{bmatrix}$$

Let $X_k = (\boldsymbol{t}, \boldsymbol{R})^{\boldsymbol{t}}$ a point where it is possible to evaluate: f(X_k), $\nabla W(X_k)$ and $\mathrm{H}(X_k)$

The minimum is reached at $X_{k+1} = X_k - \mathrm{H}(X_k)^{-1}.\nabla W(X_k)$

With $X_0 = \begin{pmatrix} 10 \\ 20 \end{pmatrix}$ as starting point.

$\forall$ X_k, det (W) = $2\pi * 0 - (2\pi * 2\pi)$ = $-4\pi^2$

$$\mathrm{H}(X_k)^{-1} = \frac{1}{-4\pi^2}\begin{bmatrix} 0 & -2\pi \\ -2\pi & 2\pi \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2\pi} \\ \frac{1}{2\pi} & -\frac{1}{2\pi} \end{bmatrix}$$

$X_{k+1} = X_k - \mathrm{H}(X_k)^{-1}.\nabla W(X_k)$
$X_1 = X_0 - \mathrm{H}(X_0)^{-1}.\nabla W(X_0)$

$$\nabla \boldsymbol{W}(\boldsymbol{X_0}) = \begin{bmatrix} \boldsymbol{\pi(2*10+2*20)} \\ \boldsymbol{2\pi*10} \end{bmatrix} = \begin{bmatrix} \boldsymbol{60\pi} \\ \boldsymbol{20\pi} \end{bmatrix}$$

$$X_1 = \begin{pmatrix} \boldsymbol{10} \\ \boldsymbol{20} \end{pmatrix} - \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\frac{1}{2\pi}} \\ \boldsymbol{\frac{1}{2\pi}} & \boldsymbol{-\frac{1}{2\pi}} \end{bmatrix} \begin{bmatrix} \boldsymbol{60\pi} \\ \boldsymbol{20\pi} \end{bmatrix} = \begin{pmatrix} \boldsymbol{10} \\ \boldsymbol{20} \end{pmatrix} - \begin{pmatrix} \boldsymbol{10} \\ \boldsymbol{20} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}$$

The minimum has been reached in only one iteration as expected, but it corresponds to the point (0,0), which is outside of the feasible region, that is the reason why Newton's methods cannot be used to solve this problem. The search method must take into consideration the constraints because they affect the position of the minimum. The most appropriate algorithm for a highly constrained problem like this case study is the heuristic Tabu search method. Tabu search was developed to ensure that a search could navigate through difficult regions of the search space such as local minima and across infeasible regions of the search space by the use of restrictions.

IX. Conclusion

The reduction of the amount of material used to build a column might be improved by optimising its cross-section area. Working on the objective function it is possible to find the stationary point(s) and consequently know the optimum point, i.e. minimum or maximum. Several search methods allow seeking optimum point by using an iterative process. However, it has been demonstrated that the choice of the kind of algorithm depends not only upon the constraints of the problem if they exist but also on the problem itself (multi-dimensional, mult-objective, linear, etc…). Here the problem is highly constrained and they impact on the position of the minimum, thus Tabu search method has been preferred since it enables to deal with a high number of boundary conditions.

X. Reference list

[1] Warren, C.Y and Richard, G.B. Roark's Formulas for Stress and Strain seventh edition. McGraw-Hill.

Appendix 1: Buckling

[1]

Formulas for long columns. The unit stress at which a long column fails by elastic instability is given by the Euler formula

$$\frac{P}{A} = \frac{C\pi^2 E}{(L/r)^2} \tag{12.1-1}$$

where P = total load, A = area of section, E = modulus of elasticity, L/r = slenderness ratio, and C is the coefficient of constraint, which depends on end conditions. For round ends, $C = 1$; for fixed ends,

Thin cylindrical tubes. For a thin cylindrical tube, the theoretical formula for the critical stress at which buckling occurs is

$$\sigma' = \frac{E}{\sqrt{3}\sqrt{1-\nu^2}} \frac{t}{R} \tag{12.2-13}$$

when R denotes the mean radius of the tube (see Table 15.2). Tests indicate that the critical stress actually developed is usually only 40–60% of this theoretical value.

Much recent work has been concerned with measuring initial imperfections in manufactured cylindrical tubes and correlating these imperfections with measured critical loads. For more detailed discussions and recommendations refer to Refs. 1–5 in this chapter and to Refs. 101–109 in Chap. 15.

Appendix 2 : Matlab script

```
% Contours
subplot(2,2,1)
[t,R] = meshgrid(0:2.5:25,0:10:100);
Z = pi*((R+t).^2-R.^2);
contourf(t,R,Z,30);
colormap(jet);
title ('Contours','fontsize',14,'fontweight','b','color','r');
xlabel('t (cm)','fontsize',12,'fontweight','b','color','b');
ylabel('R (cm)','fontsize',12,'fontweight','b','color','b');

% Surface
subplot(2,2,3)
[t,R] = meshgrid(0:2.5:25,0:10:100);
```

```
Z = pi*((R+t).^2-R.^2);
meshc(t,R,Z);
title ('Surface','fontsize',14,'fontweight','b','color','r');
xlabel('t (cm)','fontsize',12,'fontweight','b','color','b');
ylabel('R (cm)','fontsize',12,'fontweight','b','color','b');
zlabel('Area (cm^2)','fontsize',12,'fontweight','b','color','b')

% Contours & Surfaces conbined
subplot(2,2,[2 4])
[t,R] = meshgrid(-3:.025:3,-3:.1:3);
z=pi*((R+t).^2-R.^2);
meshc(t,R,z);
grid on
box on
view([130,30])
xlabel('t (cm)','fontsize',12,'fontweight','b','color','b');
ylabel('R (cm)','fontsize',12,'fontweight','b','color','b');
zlabel('Area (cm^2)','fontsize',12,'fontweight','b','color','b')
title ('Surface and contours
combined','fontsize',14,'fontweight','b','color','r');
```