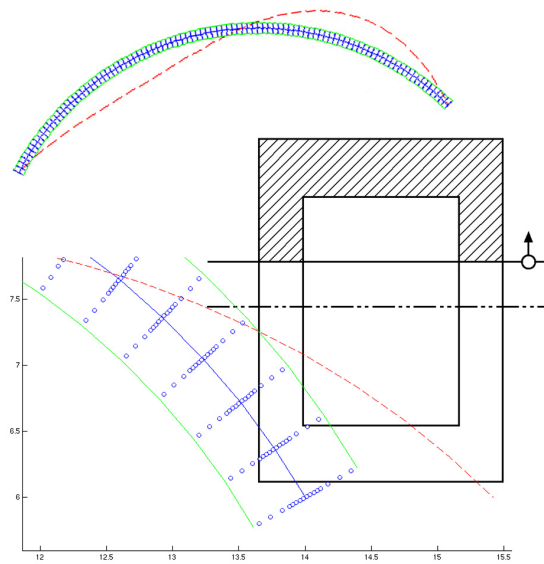




LUND
UNIVERSITY



STRESS AND STRENGTH ANALYSIS OF CURVED GLULAM BEAMS WITH BOX CROSS-SECTION

ERIK PERSSON

Structural
Mechanics

Master's Dissertation

Department of Construction Sciences
Structural Mechanics

ISRN LUTVDG/TVSM--08/5154--SE (1-93)
ISSN 0281-6679

STRESS AND STRENGTH ANALYSIS OF CURVED GLULAM BEAMS WITH BOX CROSS-SECTION

Master's Dissertation by
ERIK PERSSON

Supervisors:

Per Johan Gustafsson, Professor,
Div. of Structural Mechanics

Arne Emilsson,
Limträteknik AB, Falun

Roberto Crocetti,
Töreboda-Moelven AB, Töreboda

Examiner:

Göran Sandberg, Professor,
Div. of Structural Mechanics

Copyright © 2008 by Structural Mechanics, LTH, Sweden.
Printed by KFS I Lund AB, Lund, Sweden, August, 2008.

For information, address:
Division of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: <http://www.byggmek.lth.se>

Acknowledgements

The work presented in this master's thesis was carried out during winter and spring 2007/2008 at the Division of Structural Mechanics at Lund Institute of Technology, Lund University, Sweden.

I would like to thank my supervisor Prof. Per Johan Gustafsson for his excellent guidance throughout the work with this thesis. The extent of this thesis scope is much thanks to his knowledge in the field of structural analysis and ability to share that knowledge in a most comprehensive way.

I would also like to express my gratitude to my supervisors Arne Emilsson and Roberto Crocetti for useful help during the research.

Lund, May 2008

Erik Persson

Abstract

Glulam arches can be used when designing structures with large spans such as halls or arenas. Wood has low density compared to steel and concrete which keeps down the dead weight of the structure. The cross section of glulam arches is often built up as a box-section for efficiency reasons. However, wood shows strongly orthotropic behavior where maximum allowed stress perpendicular to grain is only 1-2% of maximum stress along grain. Bending moment creates stress perpendicular to grain for a curved beam element. By altering the slope of the arch, moment forces can be minimized for a specific load distribution and the structure can be designed using a minimum of material.

Wood is a heterogeneous material which means that different parts of the material have different properties. The volume of wood needed for designing an arch big enough for a structure as a hall or an arena is substantial. As the volume of wood under stress is increased the probability that a small part of the volume has a low ultimate stress value also increases. This effect can be handled using Weibull statistics theory.

In this thesis expressions for calculation of normal stress, shear stress and stress perpendicular to grain are derived for a curved beam with box cross section using beam theory. By using equilibrium conditions and an approximation method, the force vector corresponding to a distributed load of a curved beam element is described without the need of shape functions. With use of the finite element method, reaction forces can be calculated for an arbitrary system of curved beams. If the reaction forces are known, it's possible to calculate the section forces using the equilibrium conditions. Finally, Weibull theory is used to calculate the strength of such a structural system.

The theory is implemented using MATLAB/CALFEM to create a toolbox which can be used for strength design of structures made of curved glulam beams.

Calculations carried out using the toolkit developed show that the Weibull theory size effect has a large impact on the strength of large arches. Analysis

of an arch with a span of about 90 meters shows that the strength may be heavily overestimated if the size effect is disregarded.

Sammanfattning

Limträbågar kan användas för att bygga upp det bärande systemet i byggander med stor spännvidd som arenor eller hallar. Trä är lätt i förhållande till stål och betong vilket möjliggör en låg vikt för konstruktionen. Tvärsnittet är ofta av lådtype av effektivitetsskäl. Trä har egenskapen att vara extremt ortotrop där den maximalt tillåta spänningen vinkelrät fiberriktningen enbart är 1-2% av den maximal tillåta spänningen längs fiberriktningen. För en krökt balk är böjmoment den främsta orsaken till spänningar vinkelrät fiberriktningen. Genom att välja en bågkonstruktion med rätt krökning för en specifik last kan momentkrafterna minimeras.

Trä är ett heterogent material vilket innebär att olika delar av materialet har olika egenskaper. Den totala volymen som krävs för att konstruera en balk stor nog för en hall eller arena är substansiell. Då volymen av trä under spänning ökar, ökar också risken att någon liten del av den volymen har en lägre hållfasthet. Detta beskrivs av Weibull teori och kallas storlekseffekt.

I denna rapport har uttryck för normalspänning, skjuvspänning och spänning vinkelrät fiberriktningen härletts för en krökt balk av lådtvärsnitt med hjälp av balkteori. Genom att utnyttja jämviktsvillkor kan lastvektorn för en krökt balk under utbredd last beräknas utan att formfunktioner behöver vara kända. Med hjälp av finita element-metoden kan upplagskrafter beräknas för ett godtyckligt system av krökta balkelement. Är last och upplagskrafter kända kan snittkrafter beräknas med hjälp av jämviktsvillkor. Avslutningsvis används Weibullteori för att beräkna systemets hållfasthet.

Teorin beskriven ovan är implementerad som rutiner i MATLAB/CALFEM, dessa bygger upp en verktygslåda av funktioner som kan användas för att konstruera strukturer av godtyckliga krökta limträelement utav lådtvärsnitt.

Beräkningar utförda med hjälp av verktygslådan visar att storlekseffekten är högst påtaglig för stora bågar. Analys av en båge med drygt 90 meters spännvidd visar på att hållfastheten kraftigt överskattas om storlekseffekten inte tas med i beräkningarna.

List of Symbols

Greek letters

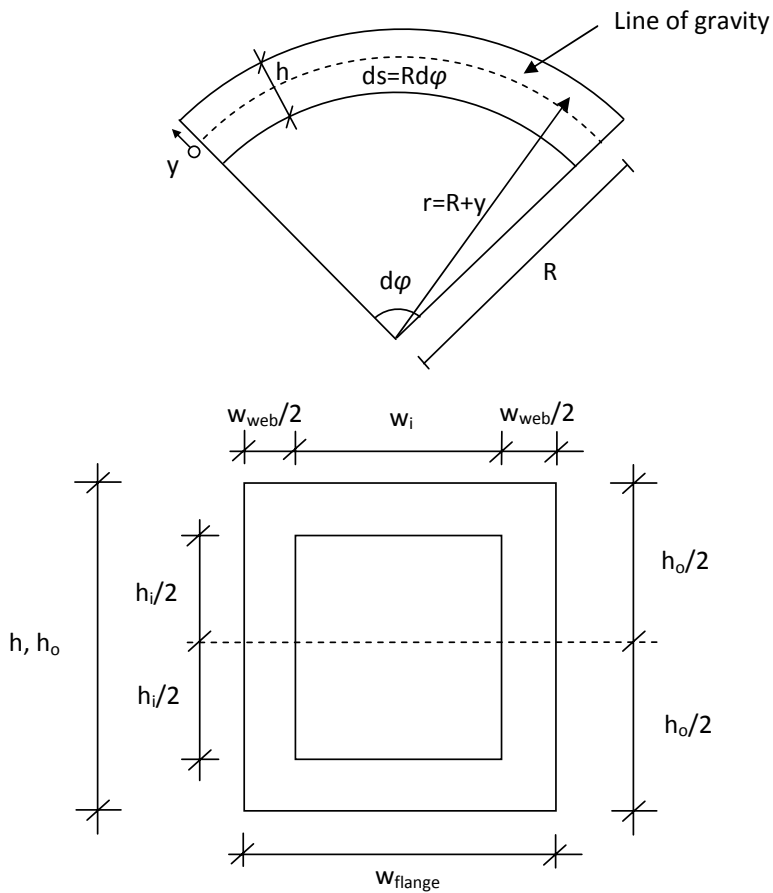
α	Angle
ε	Normal strain
γ	Shear strain
φ	Angle
θ	Angle
σ	Normal stress
σ_r	Tensile stress perpendicular to line of gravity
τ	Shear stress

Latin letters

f	Force vector
h, h _o	Total height of cross section
h _{web}	Total height of web
I	Moment of inertia
J	Moment of inertia of a curved beam
K	Stiffness matrix
m	Cross section constant
m*	Measure of spread in Weibull theory
M	Moment
N	Normal force

q	Distributed load
R	Radius of curvature
S	Probability of collapse
s	Coordinate along beam
u	Displacement vector
V	Shear force
w, b	Total width of cross section
w_{web}	Total width (thickness) of web

Definition of geometry



CONTENTS

1	Introduction	1
1.1	Background	1
1.2	Objectives.....	2
1.3	Limitations.....	3
1.4	Audience	3
2	Theory	5
2.1	Finite Element Formulation of straight and curved Beams	5
2.1.1	Element stiffness matrix	6
2.1.2	Modified moment of inertia for hollow box section	9
2.1.3	Transformation to global system	10
2.1.4	Horizontal flip of beam	11
2.2	Force vector and external load.....	13
2.2.1	Approximation of arbitrary distributed load.....	13
2.2.2	Reaction force vector.....	15
2.2.3	Force load vector	16
2.3	Section forces.....	19
2.4	Beam theory.....	21
2.4.1	Geometry	22
2.4.2	Strain	23
2.4.3	Moment and normal force	25
2.4.4	Section forces and loading	27
2.4.5	Shear force.....	29
2.4.6	Stress perpendicular to line of gravity	33
2.5	Effective stress at combined states of stress	36
2.6	Size effect and Weibull theory	36
3	Software implementation of theory	41

3.1	Output.....	41
3.2	MATLAB Functions.....	41
3.2.1	cbeam2e.....	42
3.2.2	cbeam2dl.....	42
3.2.3	cbeam2disp.....	42
3.2.4	cbeam2p.....	43
3.2.5	cbeam2stress	43
3.2.6	cbeam2s	43
3.2.7	cbeam2weibull.....	43
4	Curvature effect on stresses	45
4.1	Normal stress	46
4.2	Shear stress	47
4.3	Stress perpendicular to line of gravity	48
4.4	Comments on curvature effect on stresses	49
5	User manual – cbeam2 toolkit explained	51
6	Effect of load distribution, size and shape on strength	57
6.1	Influence of load distribution and size.....	57
6.2	Influence of shape and height.....	59
6.3	Influence of size effect.....	61
6.4	Analysis of arch 7 in Sandviken structure	62
7	Conclusions	65
7.1	Concluding remarks	65
7.2	Future work.....	65
	Bibliography	67
	Appendix A	69
	cbeam2e(ex, ey, eq, ep, flip_beam).....	69
	cbeam2dl(ex, ey, eq, ep, flip_beam, n)	72

cbeam2pdisp(ex, ey, eq, ep, flip_beam, n, ed, mfact)	74
cbeam2p(ex, ey, R, n, flip_beam)	76
cbeam2stress(ys,M,V,N,R,ep)	79
cbeam2s(ex, ey, eq, R, n, Q, flip_beam)	81
cbeam2weibull(ex, ey, es, ep, flip_beam)	83

Appendix B

CHAPTER 1

1 INTRODUCTION

1.1 BACKGROUND

Glulam timber arches can be used as an alternative to steel when designing large structural systems such as halls or arenas. The cross section is for very large arches often designed as a hollow box section to increase the bending moment capacity. Wood, as a construction material, is a renewable resource, has excellent fire withstanding abilities and has a low self weight compared to steel and concrete (Burström, 2001). An example of such a structure is the Nordic Hall, which is situated in Sundsvall, a city in north of Sweden. The span of the wooden arch building up the structural system is almost 90 meters. During winter 1993 this structure almost collapsed due to large cracks along the curved beam elements. These cracks arose because of excessively unsymmetrical snow load which created large forces perpendicular to line of gravity in the connection points of the beams building up the arch (Olsson, 2001). More of these kinds of structures are planned, during spring 2008 a glulam arch-structure with a span of 108 meters is being built in Sandviken, Sweden.

By choosing an arch as design element moment forces are minimized. The slope of an arch can be chosen in such a way that loads mainly will be transported as compressive forces attacking the ends. A good illustration of the concept is old stone arch bridges built without using any cement or steel bars. It's clear that the stone structure can't withstand more than a minimum of moment- or shear forces, the whole concept is dependent on the fact that the arch structure transfers load mainly as compressive force (Isaksson et al, 2005). The compressive force action is, however, dependant of the load

distribution. For a stone arch the dead-weight is dominating and the load carrying action is not very much affected by variations of the external load.

The shape of the arch can be chosen for a specific load so that no bending moment occurs. It is possible to derive this shape mathematically and the result, called the thrust line, is parabolic for a uniform vertical load. The reason why bending moment should be minimized is that bending moment leads to large stresses parallel to grain, stress perpendicular to grain and in general also to shear stress. Since wood shows strongly orthotropic behavior where the maximum allowed stress perpendicular to grain is only 1-2% of maximum stress along grain, stress perpendicular to grain often becomes the limiting factor when designing arches.

Wood is a heterogeneous material which means that different parts of the material have different properties. The volume of wood needed to construct arches with span as large as 90 meters is substantial. As the volume of wood under stress is increased the probability that some small part has a low ultimate stress value also increases. This is called the volume effect and can be handled using Weibull statistics, also known as weakest link theory. (Olsson, 2001)

Equations based on beam theory for calculating stresses for curved beams can be found in several collections of formulas, see e. g. Warren (2004). These equations or expressions are however generic and needs further development before application to a specific cross section. As an alternative, it would be possible to build up a complete finite element model using 3D shell elements or 2D plate elements. This would, however, require FEM-modeling skills and be very time consuming, thereby also making the strength design more expensive.

1.2 OBJECTIVES

The main objective of this master's thesis is to develop a method that can be used when designing structural systems built up by glulam arches with hollow box cross section.

The sub activities of this master's thesis are:

- Derive a method for obtaining normal force, shear force and bending moment for a structural system of curved beams using the finite element method and beam theory.
- Derive expressions for tangential normal stress, radial normal stress and shear stress for a curved glulam beam with a hollow box section.
- Suggest a method for obtaining a design value for the strength of a curved wooden beam where the volumetric effect is taken into account, using Weibull statistics.
- Implement the above derivations and methods to create a software toolbox for designing large curved beams where normal stress, shear stress and stress perpendicular to line of gravity are taken into account.

1.3 LIMITATIONS

Only stresses caused by in-plane external loading of the arch are considered. Possible torsion, out-of-plane bending and shear force, and moisture induced stresses are accordingly not considered. Neither is buckling nor possible effects of creep studied. The material is assumed to be linear elastic.

1.4 AUDIENCE

To fully comprehend this thesis, some knowledge of the finite element method, beam theory, calculus and linear algebra is recommended.

CHAPTER 2

2 THEORY

2.1 FINITE ELEMENT FORMULATION OF STRAIGHT AND CURVED BEAMS

The finite element method is used to solve partial differential equations numerically. Almost all physical phenomena are modeled using differential equations, and in most cases the equations are too complicated to be solved by classical analytical methods. The approach of the finite element method is to divide the region of interest into smaller elements, finite elements. Instead of seeking an approximation that holds for the entire region, approximation that holds for a small part of the region is used. The smaller parts are connected by their boundaries, which makes it possible to build up a global equation system describing the physical behavior of the entire region.

It is a characteristic behavior of the finite element method that as the number of finite elements used to describe a problem is increased, the error of the approximation decreases. The finite element method can be used to solve differential equations describing ground water flow, electrical current, laminar flow in pipes and many other physical engineering problems. The reference used for this chapter (2.1) is *“Introduction to the Finite Element Method”* by Ottosen and Peterson (1992).

A complete derivation of the finite element formulation for a straight beam is not carried out in this report. The interested reader can find such a derivation in e.g. Ottosen and Peterson (1992). The mechanical behavior of a straight beam can if certain assumptions are made be described by the differential equation:

$$\frac{d^2}{dx^2} \left(E \cdot I \frac{d^2 w}{dx^2} \right) - q = 0 \quad (2.1)$$

Such a beam is called a Bernoulli beam and the derivation is based on the assumption that "*plane sections normal to the beam axis remain plane and normal to the beam axis during deformation*". The critical point of this assumption is that it leads to the contradiction of existence of shear stress, τ , in spite of zero shear strain γ . The theory shows good results for long slender beams where the ratio of length to height is about six or more. For higher beams, Timoshenko beam theory which is more refined can be used. For extremely high beams, general plane stress elasticity theory has to be used. The stiffness matrices used in this paper are derived using the same basic assumptions as used in the Bernoulli beam theory for straight beams.

2.1.1 ELEMENT STIFFNESS MATRIX

The element stiffness matrix (K_e) gives the relationship between the element displacements (u_e) and the element reaction forces (f_e):

$$K_e \cdot u_e = f_e \quad (2.2)$$

The vector u_e describes displacements in the direction of the degrees of freedom and f is the corresponding force vector.



Figure 2.1. Straight beam element

The stiffness matrix of a 2D straight Bernoulli beam element can e.g. be found in (Austrell et al, 2004) and is as follows:

$$K_e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (2.3)$$

L denotes length of the beam, E denotes Young's modulus. A and I denotes area and moment of inertia of the beam cross section, respectively.

The stiffness matrix of a curved beam is more complicated because of more coupling. For instance is displacement u_4 for a curved beam (see figure 2.1) coupled not only to the forces f_1 and f_4 , but also coupled to f_3 and f_6 .

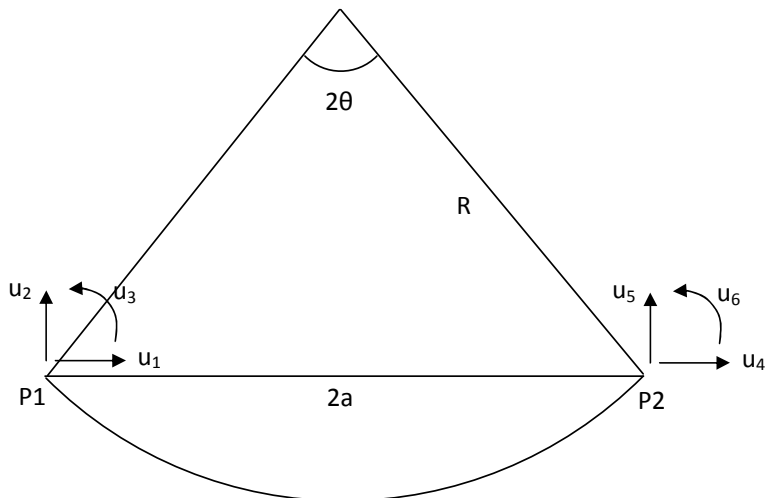


Figure 2.2. Curved beam element

S. Krenk (1993) has developed a general method to compute the stiffness matrix for curved and non-homogenous beam elements. The stiffness matrix is obtained using a number of stress energy terms. The length a and angle θ are defined in figure 2.2. The stiffness matrix K_e for a curved beam with constant radius R , in its local coordinates, given by:

$$K_e = G^T D G \quad (2.4)$$

Where G and D are defined in (2.5) and (2.6).

$$G = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -a & 0 & 0 & a \\ 0 & 1 & a & 0 & -1 & a \end{bmatrix} \quad (2.5)$$

$$D = \begin{bmatrix} H_{22}/H & -H_{12}/H & 0 \\ -H_{12}/H & H_{11}/H & 0 \\ 0 & 0 & 1/H_{33} \end{bmatrix} \quad (2.6)$$

where

$$\begin{aligned} H_{11} &= 2 \left(\frac{R}{EA} + \frac{R^3}{EJ} \right) \theta \cos^2 \theta + \frac{R^3}{EJ} (\theta - 3 \sin \theta \cos \theta) \\ H_{22} &= 2 \left(\frac{R}{EA} + \frac{R^3}{EJ} \right) \theta \sin^2 \theta \\ H_{12} &= 2 \left(\frac{R}{EA} + \frac{R^3}{EJ} \right) \theta \sin \theta \cos \theta - 2 \frac{R^3}{EJ} \sin^2 \theta \\ H_{33} &= \frac{R^3}{EJ} (\theta - \sin \theta \cos \theta) \\ H &= H_{11} H_{22} - H_{12}^2 \end{aligned} \quad (2.7)$$

J denotes a cross-section parameter which for a straight beam corresponds to the moment of inertia, I :

$$J = \int_A \frac{(r-R)^2}{r/R} dA \quad (2.8)$$

Where R denotes the radius of the centroid line and r is an arbitrary point located at the distance r from the centre of the circle, see figure 2.3. (Bresler and Lin, 1960)

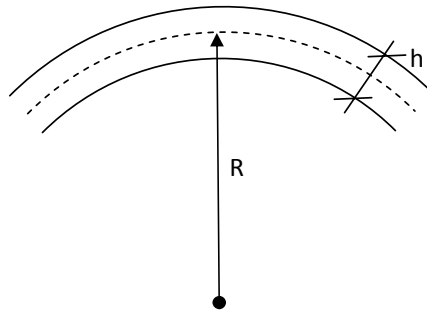


Figure 2.3. Geometry of a curved beam element

2.1.2 MODIFIED MOMENT OF INERTIA FOR HOLLOW BOX SECTION

J can, for a solid rectangular cross section, be calculated by equation (2.9) where w denotes the width and h denotes height (Bresler and Lin, 1960).

$$J = R^3 w \ln\left(\frac{2R+h}{2R-h}\right) - R^2 wh \quad (2.9)$$

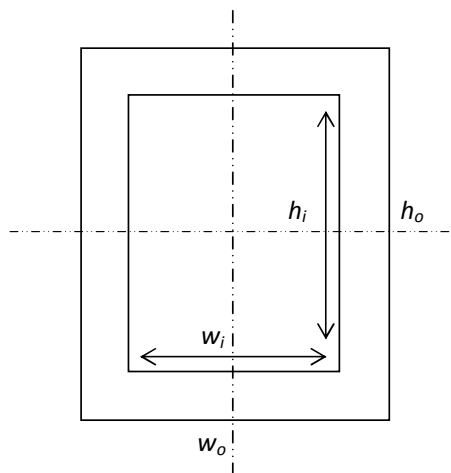


Figure 2.4. Hollow box section

The modified moment of inertia for a hollow double-symmetrical box cross-section of the type shown in figure 2.4 can be obtained as the difference between the integral of the whole section and the integral of the inner section. With parameters according to figure 2.4., equation (2.9) becomes for a hollow box section:

$$\begin{aligned}
 J_{box} &= \int_A \frac{(r-R)^2}{r/R} dA - \int_{A_{hollow}} \frac{(r-R)^2}{r/R} dA \left[\begin{array}{l} A = w \cdot h \\ dA = w \cdot dy \\ y = r - R \end{array} \right] = \\
 &= w_o \int_{-h_o/2}^{h_o/2} \frac{Ry^2}{y+R} dy - w_i \int_{-h_i/2}^{h_i/2} \frac{Ry^2}{y+R} dy
 \end{aligned} \tag{2.10}$$

If the integrals are expanded and the expression condensed, equation (2.10) will take the form:

$$J_{box} = R^3 \left[w_o \ln \left(\frac{2R+h_o}{2R-h_o} \right) - w_i \ln \left(\frac{2R+h_i}{2R-h_i} \right) \right] - R^2 (w_o h_o - w_i h_i) \tag{2.11}$$

Where w_o and h_o denotes outer width and height and w_i and h_i width and height of the hollow section. Note that equation (2.11) is valid only if the cross section is double-symmetrical.

2.1.3 TRANSFORMATION TO GLOBAL SYSTEM

The stiffness matrix given by (2.4) is valid only as long as the orientation of the degrees of freedom is equal to the axis of the coordinate system. To be able to assemble a system of multiple beam elements with different orientations a transformation (rotation) needs to be done. This can be achieved with the use of a transformation matrix, which holds information of the orientation of the individual beam. The transformation is carried out using equation (2.12). The equation can be found in Austrell, et. al. (2004).

$$K_e^g = A^T K_e A \tag{2.12}$$

The transformation matrix A can be calculated by different methods, see for example Calfem – a finite element toolbox (Austrell et al, 2004) where the following result is presented:

$$A = \begin{bmatrix} n_{x\bar{x}} & n_{y\bar{x}} & 0 & 0 & 0 & 0 \\ n_{x\bar{y}} & n_{y\bar{y}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_{x\bar{x}} & n_{y\bar{x}} & 0 \\ 0 & 0 & 0 & n_{x\bar{y}} & n_{y\bar{y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_{x\bar{x}} = n_{y\bar{y}} = \frac{x_2 - x_1}{L} \quad n_{y\bar{x}} = -n_{x\bar{y}} = \frac{y_2 - y_1}{L} \quad (2.13)$$

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Where (x_1, y_1) and (x_2, y_2) denotes the coordinates from start point to the end point of the beam in the global coordinate system.

2.1.4 HORIZONTAL FLIP OF BEAM

It's of importance to be aware of the definition of direction of curvature of curved beams. The stiffness matrix given in chapter 2.1.1 and shown in figure 2.2 can be said to have a negative curvature in the local system. Let α^* denote the angle from $\overline{P_1P_2}$ to the tangent of the beam in P1, see figure 2.5. If the angle α^* is less than or equal to π , the curvature is positive.

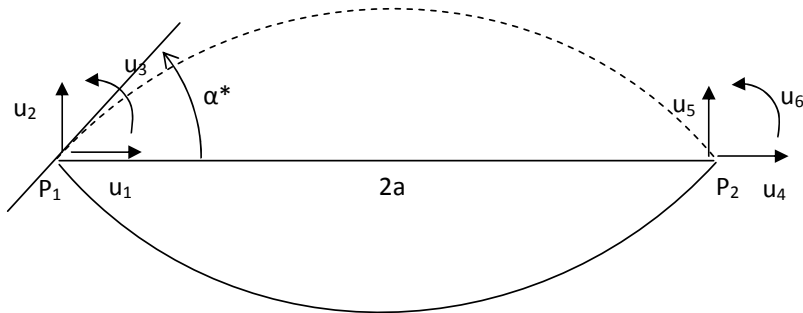


Figure 2.5. Horizontally flipped curvature of beam. Solid line: negative curvature, dotted line: positive curvature.

Since this paper focuses on arches that normally have the opposite curvature a default transformation method would be handy. It is of course possible to

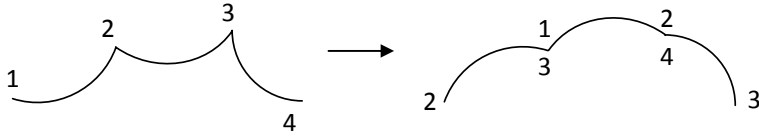


Figure 2.6. Effect of element 180 degree rotation

rotate the beam 180 degrees using the transformation matrix discussed in chapter 2.1.3. This does however have the effect that the groups of degrees of freedom locally switch places. When working with multiple beam elements that are connected this complicates the global assembly since the local DOF numbering no longer directly translates to the global numbering. This is illustrated in figure 2.6 where beam elements with two degrees of freedom each are rotated.

One solution is to number the elements in such a way that all elements are rotated 180 degrees as shown on the right side of figure 2.6. This is however not a very intuitive method. A better solution is to "horizontally flip" the stiffness matrix with respect to rotations. If the stiffness matrix is element-wise multiplied with a "horizontal flip" matrix, the stiffness matrix of an equivalent beam with opposite curvature is obtained. Such a matrix can be created by consideration to the effects of the displacement of two curved beam elements, mirrored around the x-axis as shown in figure 2.5. If equation (2.2) is used, all displacements are set to zero except degree of freedom 1. Then the reaction forces are calculated for both beams. If there is a sign difference, a minus sign will go to the corresponding place in the "flip"-matrix. The same methodology is then carried out for all degrees of freedom to complete the "flip"-matrix:

$$A_{hf} = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix} \quad (2.14)$$

2.2 FORCE VECTOR AND EXTERNAL LOAD

2.2.1 APPROXIMATION OF ARBITRARY DISTRIBUTED LOAD

When designing structural systems the major design criterion is normally the ability to withstand external load. Other design criteria can e.g. be sound isolation ability, the capacity to withstand fire or long-term moisture exposure reactions. (Isaksson et al, 2005)

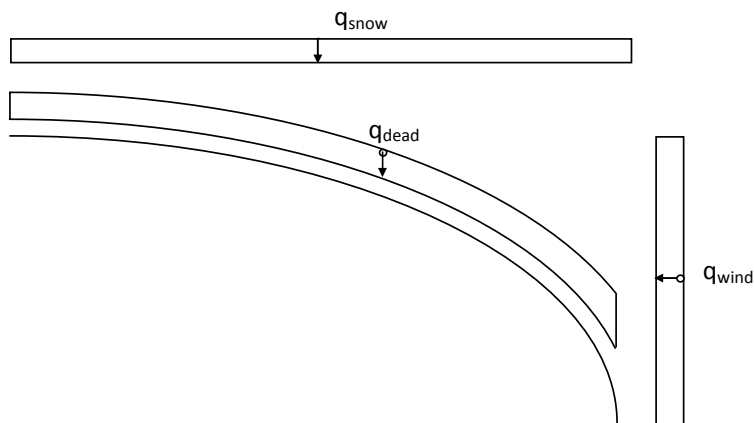


Figure 2.7. Different examples of load

The main loads when designing an arch that will carry a roof are:

- Snow load
- Wind load
- Dead weight

Figure 2.7 shows an example of how the different distributed loads can attack. The dead weight load could be approximated as a number of triangular shaped distributed loads applied in vertical direction.

To describe an arbitrary distributed load, a corresponding finite element load vector valid for curved beam elements is needed.

An arbitrary distributed load can be approximated as a number of point loads as in figure 2.8. This will not give the exact solution but the error reduces as

the number of point loads is increased. To compute the load vector that corresponds to an arbitrary located point load, the shape functions of the beam element must be known. Shape functions for a curved beam become quite complex. Litewka and Rakowski (1996) suggest an approximation based on expansion of trigonometric functions in the power series. These kinds of shape functions become very unhandy and complex.

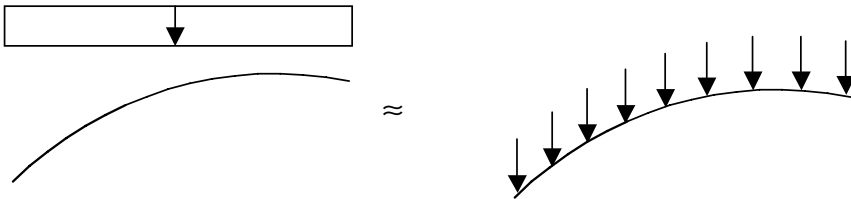


Figure 2.8. Approximation of distributed load

To avoid complex shape functions the beam element can be divided into smaller elements where the approximate point load attacks directly into the nodes. If there are no load on the span of the (small) element the force vector doesn't depend on shape functions for calculation of the equivalent load. Instead, the load can be put directly into the corresponding degree of freedom of the force vector. The orientation of the force does of course have to match the directions of the global coordinate system. Such a transformation is easy to achieve, the force of arbitrary direction just has to be divided into two components parallel with the global x-axis and y-axis.

The drawback of such a solution with short elements is that it takes more time to compute. On the upside all possible kinds of distributed loads can be approximated without restrains.

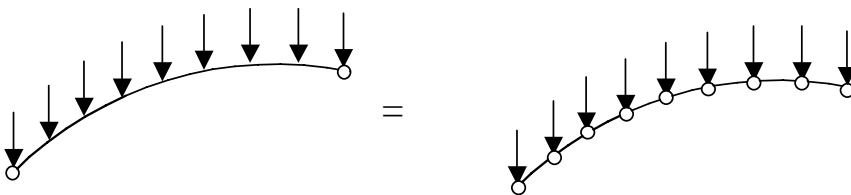


Figure 2.9. One element divided into many parts.

The method chosen here to calculate the load vector for a curved beam element is to approximate an arbitrary distributed load as a number of equivalent point loads, and then divide the curved beam element into shorter elements matching the number of point loads as shown in figure 2.8 and 2.9. As the system is solved the load vector can be obtained, see chapter 2.2.3.

2.2.2 REACTION FORCE VECTOR

By studying the equilibrium conditions of a small beam element the load vector can be derived. See figure 2.10 which represents a small part of a beam. A reasonable approximation for a small part of the beam is to divide the distributed load equally into the nodes in vertical and horizontal direction. If the distributed load is divided equally into each nodal point then $R_{Ay} = R_{By}$ and $R_{Ax} = R_{Bx}$:

$$R_{Ay} = R_{By} = -\frac{q_y(x_2 - x_1)}{2} \quad (2.15)$$

$$R_{Ax} = R_{Bx} = -\frac{q_x(y_2 - y_1)}{2} \quad (2.16)$$

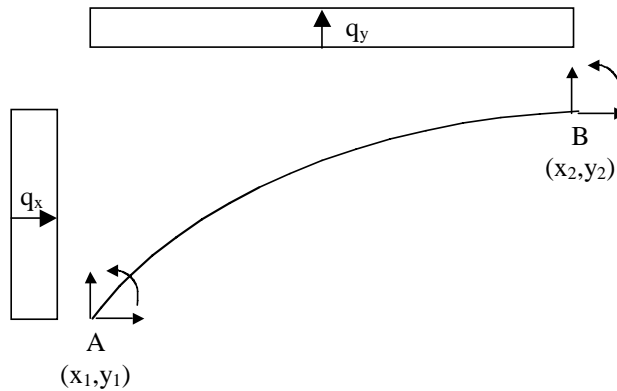


Figure 2.10. Equilibrium conditions.

With the above assumption, moment reaction forces in both ends will be zero:

$$-q_y \frac{(y_2 - y_1)^2}{2} + q_x \frac{(x_2 - x_1)^2}{2} + R_{By}(x_2 - x_1) - R_{Bx}(y_2 - y_1) + M_B = 0 \quad (2.17)$$

if (2.19) is rearranged we get M_B (2.20)

$$M_B = q_y \frac{(y_2 - y_1)^2}{2} - q_x \frac{(x_2 - x_1)^2}{2} - R_{By}(x_2 - x_1) + R_{Bx}(y_2 - y_1) = 0 \quad (2.18)$$

and with point B as base the moment equilibrium is described by equation (2.19):

$$q_y \frac{(y_2 - y_1)^2}{2} - q_x \frac{(x_2 - x_1)^2}{2} + R_{Ay}(x_2 - x_1) - R_{Ax}(y_2 - y_1) + M_A = 0 \quad (2.19)$$

which if rearranged gives M_A

$$M_A = -q_y \frac{(y_2 - y_1)^2}{2} + q_x \frac{(x_2 - x_1)^2}{2} - R_{Ay}(x_2 - x_1) + R_{Ax}(y_2 - y_1) = 0 \quad (2.20)$$

When all reaction forces are known, the reaction force vector can be described using by presenting equation (2.15 - 2.20) in matrix notation:

$$f_r = - \begin{bmatrix} (x_2 - x_1)/2 \cdot q_y \\ (y_2 - y_1)/2 \cdot q_x \\ 0 \\ (x_2 - x_1)/2 \cdot q_y \\ (y_2 - y_1)/2 \cdot q_x \\ 0 \end{bmatrix} \quad (2.21)$$

2.2.3 FORCE LOAD VECTOR

Using the previously described method to approximate an arbitrary distributed load the reaction forces of an element can be calculated. The solution of the system of equations developed is based on the condition that the structural system is in equilibrium state. This implies that the reaction forces of a beam element are equivalent but negative to the external force, which means that the reaction forces can be used to describe the external load. To avoid complications, a section evaluated should not spread over more than one quarter of a circle. See chapter 2.3 for further discussion regarding this matter.

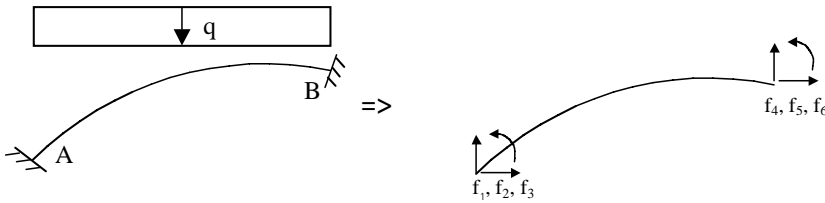


Figure 2.11. Reaction forces.

If displacements of the beam endings are restrained, reaction forces can be developed. For a beam element, the reaction forces f_b^e plus the external load f_l^e equals the stiffness time displacement as described by equation (2.22) (Ottosen et al, 1992).

$$K^e a^e = f_b^e + f_l^e \quad (2.22)$$

If displacements, a^e , is set to zero we get

$$K^e \cdot [0] = f_b^e + f_l^e \quad (2.23)$$

$$f_l^e = -f_b^e \quad (2.24)$$

$$f_l = - \begin{bmatrix} R_A \\ R_B \end{bmatrix} \quad (2.25)$$

This means that the force vector now can be calculated (approximated) without using shape functions. The tools to describe the stiffness matrix of a curved box-section beam element and the equivalent force vector of arbitrary distributed load acting on such a beam is now derived. This makes it possible to describe a complex arch where e.g. load and radius varies. To summarize the process:

- The distributed load is approximated as a number of point loads (figure 2.8)
- The curved beam element is divided into a sub-system of smaller curved beam elements so that the each point-load corresponds to a connection point between two sub elements (figure 2.9)
- The end displacements are set to zero and the system is solved (figure 2.11)

- The reaction forces can be used to describe the corresponding load vector (equation 2.25)

The reason to calculate the corresponding force vector instead of settling for the system of many elements is compatibility. If both the stiffness matrix and force vector are known, the curved beam element can be a part of a system consisting of multiple types of elements in a much more easy-to-handle way. This approach will make it easy to combine bar elements, straight beam elements and curved beam elements using CALFEM and the CBEAM2-toolkit.

2.3 SECTION FORCES

If reaction forces and external loads are known for a beam element, it's possible to evaluate the section forces in all points using equilibrium condition for the subsystem of interest. See e.g. (Heyden, 2005).

P' is the point where normal force, shear force and moment are to be evaluated. Equilibrium gives expressions for N' , V' and M' , see figure 2.12.

$$\rightarrow R_x + q_x \cdot l'_x + N' = 0 \quad (2.26)$$

$$\uparrow R_y + q'_y \cdot l'_x + V' = 0 \quad (2.27)$$

$$\curvearrowright M_R - R_y \cdot l'_x + R_x \cdot l'_y - q'_y \cdot \frac{l'^2_x}{2} + q'_x \cdot \frac{l'^2_y}{2} + M' = 0 \quad (2.28)$$

Note that (2.28) only is definitely true as long as P' and P_R are in the same quarter of the unit circle if external load exists. This condition is easy to fulfill by using a reasonable element mesh.

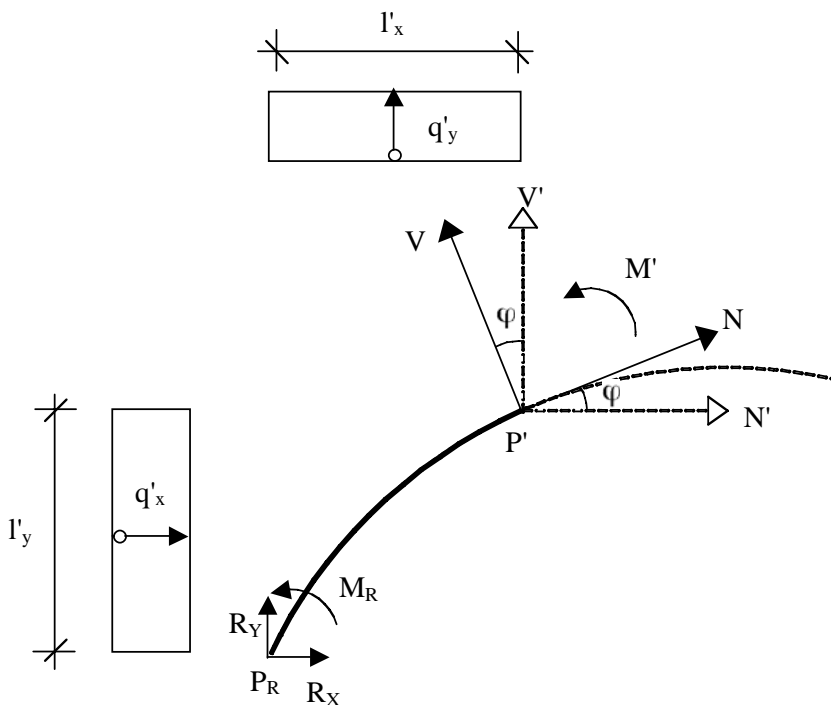


Figure 2.12. Evaluation of section forces.

It's now just a matter of transforming V' and N' to finally derive V and N . Basic trigonometry gives shear force V according to:

$$V = \cos \varphi \cdot V' - \sin \varphi \cdot N' \quad (2.29)$$

In the same way, the normal force N is determined using equation (2.32)

$$N = \cos \varphi \cdot N' + \sin \varphi \cdot V' \quad (2.30)$$

If equation (2.30) is rearranged, M' can be derived. Since moment is a scalar rather than a vector, no transformation needs to be done.

$$M = M' = -M_R + Ry \cdot l'x - Rx \cdot l'y + q'y \cdot \frac{l_x^2}{2} - q'x \cdot \frac{l_y^2}{2} \quad (2.31)$$

Careful consideration needs to be taken when section forces are evaluated for a curved beam element that reaches over more than one quarter of the unit circle if external loads exist. This is due to that fact the distributed loads q_y and q_x may individually give both positive and negative contributions to M' if the point of evaluation is within certain intervals. This not is taken into consideration in the above derivation of the section forces.

2.4 BEAM THEORY

In the following chapter expressions for shear stress and tension stress perpendicular to grains are derived. The derivation presented here is based on professor Gustafsson's notes made for a previous more comprehensive version of the course "Beam Theory" (code VSM091) given by the Division of Structural Mechanics. The author of this thesis has derived the stress equations from general form to the specific case of a hollow box section. The shear force (chapter 2.4.5) is derived by using a different approach than used in Gustafsson's lecture notes.

The following assumptions are made

- Displacements and strains are small
- The material is linear elastic
- Cross section is double-symmetrical
- A plane cross section will stay plane during loading
- The cross section will stay perpendicular to the centre line of the beam
- The external load is applied distributed on the cross-section and with such a distribution that the shear stress and the stress perpendicular to grain are affected only by N , V and M , and not by local load distribution.

Even though there is no shear strain due to the assumption that the cross-section remain perpendicular to the centre-line, shear stresses must exist due to equilibrium conditions. Both the shear stress and stress perpendicular to grain are derived by equilibrium conditions, not by strain calculation and use of the stress-strain relation.

2.4.1 GEOMETRY

First, the geometry of an infinitely short curved beam element needs to be defined.

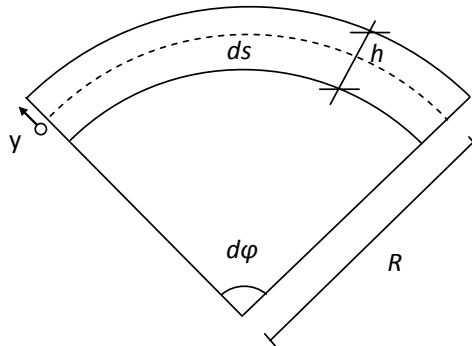


Figure 2.13. Definition of geometry of an infinitely short beam element, ds .

Sinus for an angle $d\varphi$ can be approximated as the angle itself and cosine for $d\varphi$ can be approximated as 1 ($\cos(0) = 1$). The length ds is related to $d\varphi$ by:

$$ds = R d\varphi \quad (2.32)$$

Let r be a coordinate such that $r = (R+y)$, then the length $r d\varphi$ is:

$$r d\varphi = (R + y) d\varphi = (R + y) \frac{ds}{R} \quad (2.33)$$

Displacements are described by the variables $v(s)$ and $u(s,y)$ where $v(s)$ is the radial displacement and $u(s,y)$ denotes the tangential displacement. s is the coordinate along the centre line of the beam. See figure 2.14 on the next page for definitions.

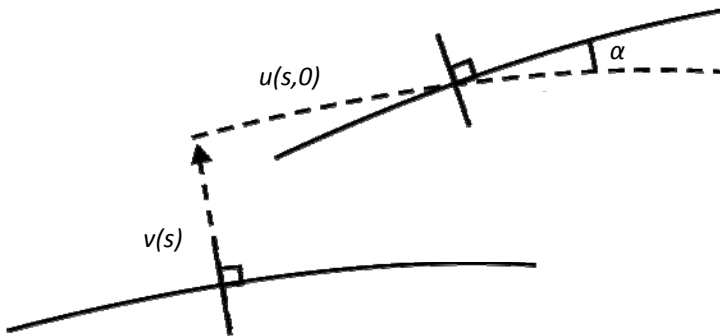


Figure 2.14. Definition of displacement, with location of centre line (line of gravity) before and after loading

Since sinus for small angles can be approximated as the angle itself, the displacement angle α can be calculated as:

$$\alpha \approx \frac{dv}{ds} \quad (2.34)$$

The tangential displacement u at height y on the cross section can be expressed as (2.37)

$$u(s, y) = u(s, 0) - y \frac{dv}{ds} \quad (2.35)$$

2.4.2 STRAIN

Normal strain is defined as the change of length over undeformed length (2.38)

$$\varepsilon = \frac{\Delta l}{l} \quad (2.36)$$

Since the undeformed length is $rd\varphi$, the strain can be calculated as:

$$\varepsilon(s, y) = \frac{u(s + ds, y) - u(s, y)}{rd\varphi} \quad (2.37)$$

Using (2.33), (2.37) can be expressed as:

$$\frac{u(s+ds, y) - u(s, y)}{rd\varphi} = \frac{R}{R+y} \left(\frac{u(s+ds, y) - u(s, y)}{ds} \right) \quad (2.38)$$

Using the fundamental theorem of calculus, (2.38) can be rewritten as (2.39)

$$\frac{R}{R+y} \left(\frac{u(s+ds, y) - u(s, y)}{ds} \right) = \frac{R}{R+y} \left(\frac{du(s, y)}{ds} \right) \quad (2.39)$$

If (2.37) is inserted into (2.41), we get

$$\frac{R}{R+y} \left(\frac{du(s, y)}{ds} \right) = \frac{R}{R+y} \left(\frac{du(s, 0)}{ds} - y \frac{d\left(\frac{dv}{ds}\right)}{ds} \right) = \quad (2.40)$$

$$= \frac{R}{R+y} (u'_0 - yv'') = \frac{R}{R+y} \left(u'_0 + \frac{y}{R} u'_0 - \frac{y}{R} u'_0 - yv'' \right) = \quad (2.41)$$

$$= \frac{Ru'_0}{R+y} + \frac{Ryu'_0}{(R+y)R} - \frac{Ryu'_0}{(R+y)R} - \frac{Ryv''}{R+y} = \quad (2.42)$$

Which finally gives an expression for the strain at height y :

$$= u'_0 - \frac{y}{R+y} (u'_0 + Rv'') = \varepsilon(s, y) \quad (2.43)$$

(2.43) can be expressed as (2.44) using matrix notation

$$\varepsilon(s, y) = \begin{bmatrix} 1 & \frac{-y}{R+y} \end{bmatrix} \begin{bmatrix} u'_0 \\ u'_0 + Rv'' \end{bmatrix} \quad (2.44)$$

It is easy to realize that u'_0 denotes relative change of length, strain, at the line of gravity. The variable v'' contains the change of curvature of the beam. New notations are introduced for u'_0 and v'' :

$$\begin{bmatrix} u'_0 \\ v'' \end{bmatrix} = \begin{bmatrix} \varepsilon_0 \\ \kappa \end{bmatrix} \quad (2.45)$$

Using (2.44) and (2.45), the strain can be expressed as

$$\varepsilon = \frac{R}{R+y} \begin{bmatrix} 1 & -y \\ & \kappa \end{bmatrix} \begin{bmatrix} \varepsilon_0 \\ \kappa \end{bmatrix} \quad (2.46)$$

Hooke's law states that the stress is Young's modulus times the strain:

$$\sigma = \varepsilon \cdot E \quad (2.47)$$

If (2.49) is inserted into (2.48), a stress equation is obtained (2.50):

$$\sigma = \frac{ER}{R+y} \begin{bmatrix} 1 & -y \\ & \kappa \end{bmatrix} \begin{bmatrix} \varepsilon_0 \\ \kappa \end{bmatrix} \quad (2.48)$$

2.4.3 MOMENT AND NORMAL FORCE

The normal force is defined as the integral (2.49) and moment is defined as the integral (2.50), see e.g. Heyden et al (2005).

$$N = \int_A \sigma dA \quad (2.49)$$

$$M = \int_A -y \sigma dA \quad (2.50)$$

With (2.48) inserted into (2.49) and (2.50) we get

$$N = \int_A \sigma dA = E\varepsilon_0 \int_A dA + E(\varepsilon_0 + R\kappa) \int_A \frac{-y}{R+y} dA \quad (2.51)$$

$$M = \int_A -y \sigma dA = -E\varepsilon_0 \int_A y dA + E(\varepsilon_0 + R\kappa) \int_A \frac{-y^2}{R+y} dA \quad (2.52)$$

Let m denote the integral (2.53). The value of m depends only on the geometry of the cross section of the beam. Hence; if the cross section is constant along the beam then the parameter m also will be constant.

$$m = \frac{1}{A} \int_A \frac{-y}{R+y} dA \quad (2.53)$$

Using m , (2.51) and (2.52) can be rewritten in a more compact form.

$$N = \int_A \sigma dA = E\varepsilon_0 A + E(\varepsilon_0 + R\kappa) Am \quad (2.54)$$

$$M = \int_A -y\sigma dA = -E\varepsilon_0 0 + E(\varepsilon_0 + R\kappa)RAm \quad (2.55)$$

Using matrix notation, (2.54) and (2.55) can be expressed as:

$$\begin{bmatrix} N \\ M \end{bmatrix} = \begin{bmatrix} EA & EAm \\ 0 & ERAm \end{bmatrix} \begin{bmatrix} \varepsilon_0 \\ \varepsilon_0 + R\kappa \end{bmatrix} = EA \begin{bmatrix} 1+m & mR \\ mR & mR^2 \end{bmatrix} \begin{bmatrix} \varepsilon_0 \\ \kappa \end{bmatrix} \quad (2.56)$$

In order to express stress using normal force and moment (2.56) needs to be inverted:

$$\begin{bmatrix} \varepsilon_0 \\ \kappa \end{bmatrix} = \frac{1}{EAR^2} \begin{bmatrix} R^2 & -R \\ -R & \frac{1+m}{m} \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} \quad (2.57)$$

If (2.57) is inserted into (2.48), an expression to calculate the tangential stress σ is finally obtained:

$$\sigma = \frac{ER}{R+y} \begin{bmatrix} 1 & -y \end{bmatrix} \frac{1}{EAR^2} \begin{bmatrix} R^2 & -R \\ -R & \frac{1+m}{m} \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} = \quad (2.58)$$

$$= \begin{bmatrix} \frac{1}{A} & -\frac{1}{RA} \left(1 + \frac{y}{m(R+y)} \right) \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix} = \frac{N}{A} - \frac{M}{RA} \left(1 + \frac{y}{m(R+y)} \right) \quad (2.59)$$

2.4.4 SECTION FORCES AND LOADING

In order to evaluate the connection between loading q_r and q_t and section forces N , V and M an infinitely small piece of the curved beam is studied using the equilibrium conditions.

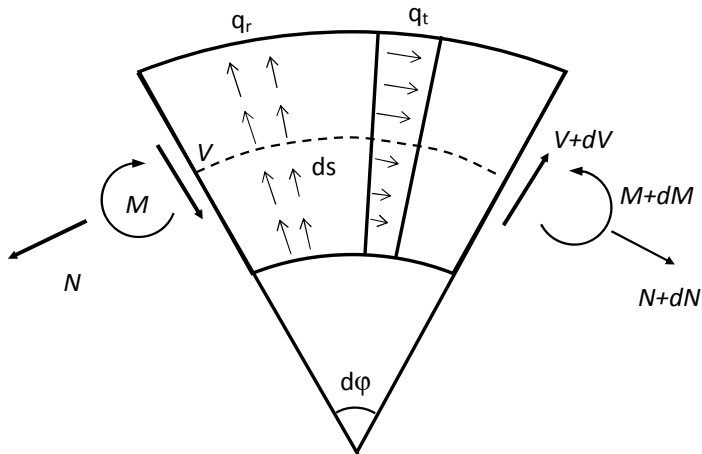


Figure 2.15. Section forces and load.

Moment equilibrium around the center of the left side of the arch gives the correlation between moment and normal force:

$$\sum M - (M + dM) - (V + dV)ds - \int_A q_r dA ds \frac{ds}{2} = 0 \quad (2.60)$$

(2.60) is rearranged:

$$-dM - Vds - \underbrace{dVds}_{\approx 0} - \int_A q_r dA \underbrace{ds^2}_{\approx 0} \frac{1}{2} = 0 \quad (2.61)$$

Which makes it possible to eliminate some terms and (2.62) is condensed to:

$$-\frac{dM}{ds} = V \quad (2.62)$$

Equilibrium in vertical (parallel to left side of cross section) direction:

$$\sum V - V + (V + dV) + \int_A q_r dA ds - d\phi(N + dN) = 0 \quad (2.63)$$

(2.63) is rearranged:

$$dV + \int_A q_r dA ds - d\phi N + \underbrace{d\phi dN}_{=0} = 0 \quad (2.64)$$

And since $d\phi = \frac{ds}{R}$, (2.64) can be rewritten as:

$$dV - \frac{ds}{R} N + \int_A q_r dA ds = 0 \quad (2.65)$$

Equilibrium in horizontal direction (perpendicular to left side cross section):

$$N - (N + dN) - d\phi(V + dV) - \int_A q_t dA ds = 0 \quad (2.66)$$

(2.66) is rearranged:

$$-dN - d\phi V - \underbrace{d\phi dV}_{=0} - \int_A q_t dA ds = 0 \quad (2.67)$$

Some terms can be eliminated:

$$-\frac{dN}{ds} - \frac{ds}{R} \frac{V}{ds} - \int_A q_t dA \frac{ds}{ds} = 0 \quad (2.68)$$

(2.68) rewritten using a more compact fashion:

$$-N' - \frac{V}{R} - \int_A q_t dA = 0 \quad (2.69)$$

Finally, the volume load can be expressed in terms of moment, normal force and shear force.

$$\int_A q_r dA = \frac{N}{R} - V' = \frac{N}{R} + M'' \quad (2.70)$$

$$\int_A q_t dA = -\frac{V}{R} - N' = \frac{M'}{R} - N' \quad (2.71)$$

2.4.5 SHEAR FORCE

In order to be able to derive an expression for shear stress parallel to line of gravity, a small portion of the beam is studied in equilibrium state. The external load (q_r , q_t) is neglected in the derivation. This approach makes the derived equation valid only where no external load attacks unless the load is applied in such a way that it does not affect the shear force or stress perpendicular to center of gravity.

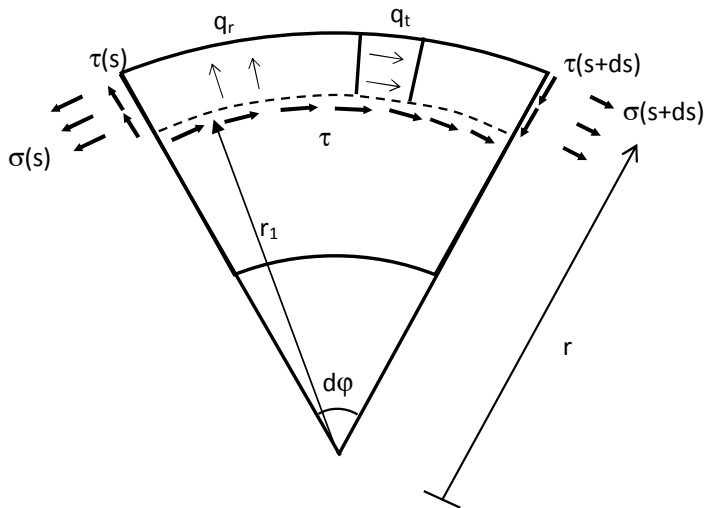


Figure 2.16. Moment equilibrium.

The same notation as earlier is used. Let r_1 denote where on the cross section the shear force is to be evaluated and $(R + h/2)$ the maximum reach of the cross section. The area made up by the cross section from r_1 to $(R + h/2)$ is denoted A_s . w denotes the width of the cross section, h denotes cross section height, see figure 2.17 for definitions.

$$r = R + y \quad (2.72)$$

$$r_1 = R + y_{start}$$

Moment equilibrium around the center of the circle of the curved beam is studied:

$$\int_{A_s} \sigma(s) r dA - \int_{A_s} \sigma(s + ds) r dA - \tau w r_1 d\phi r_1 = 0 \quad (2.73)$$

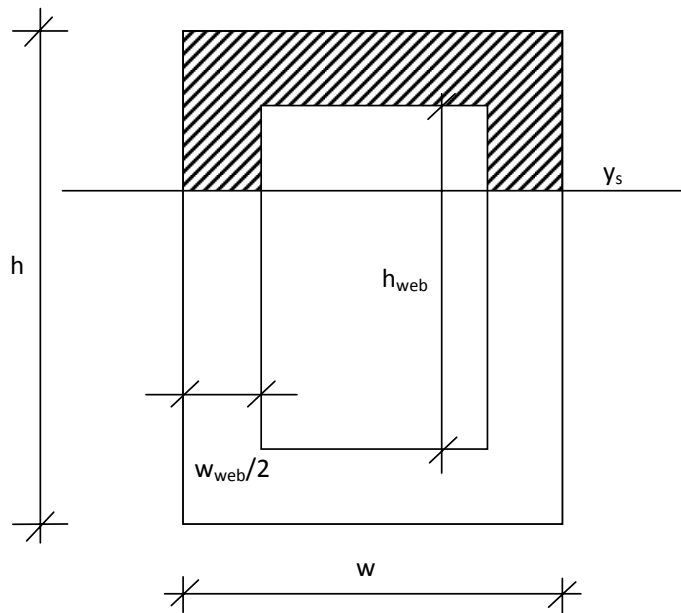


Figure 2.17. Geometric definitions of cross section.

(2.73) is rearranged using (2.33) and divided by ds :

$$\int_{A_s} \sigma(s) r dA - \int_{A_s} \frac{\sigma(s+ds) - \sigma(s)}{ds} r dA - \tau w r_1^2 \frac{ds}{R ds} = 0$$

(2.74)

Using the fundamental theorem of calculus, (2.74) becomes:

$$\int_{A_s} \sigma'(s) r dA + \tau \frac{w(R+y_s)^2}{R} = 0 \quad (2.75)$$

Solving this equation with respect to τ , a general equation for shear stress is finally obtained:

$$\tau = \frac{R}{w(R+y_s)^2} \int_{A_s} \sigma'(s) (R+y) dA = 0$$

$$(2.76)$$

If the integral above is expanded, a direct equation for shear stress is obtained. This integral does obviously depend of the shape of the cross section. For a hollow box section the equation needs to be divided into three cases.

The derivative of equation (2.59) with respect to ds yields as follows:

$$\frac{d\sigma}{ds} = \frac{N'}{A} - \frac{M'}{RA} \left(1 + \frac{y}{m(R+y)} \right) \quad (2.77)$$

Since the influence of local tangential load is neglected, equation (2.71) becomes:

$$N' = \frac{V}{R} \quad (2.78)$$

Inserting (2.78) into (2.77), and then using (2.62), some terms can be eliminated.

$$\sigma' = \frac{V}{RA} - \frac{V}{RA} \left(1 + \frac{y}{m(R+y)} \right) = \frac{-Vy}{ARm(R+y)} \quad (2.79)$$

Then if (2.76) and (2.79) are combined, equation (2.80) gives a general formula for the shear stress.

$$\tau = \frac{V}{w_{ys}(R+y_{ys})^2 Am} \int_{As} y dA \quad (2.80)$$

Suppose constant width over a specific area, then:

$$dA = w dy \quad (2.81)$$

For a cross section of constant width, (2.81) can be inserted into (2.80), giving:

$$\tau = \frac{V}{w_{ys}(R+y_{ys})^2 Am} \int_{ys}^{h/2} w_{ys} y dy \quad (2.82)$$

For a hollow box section, three equations need to be derived; one for the upper flange, one valid for the web and one valid for the lower flange. The shear stress in the upper flange can be calculated from

$$\tau_{uf} = \frac{V}{w_f(R+y_s)^2 Am} \int_{y_s}^{h/2} w_f y \, dy \quad (2.83)$$

$$\tau_{uf} = \frac{V}{w_f(R+y_s)^2 Am} W_f \left[\frac{y^2}{2} \right]_{y_s}^{h/2} \quad (2.84)$$

$$\tau_{uf} = \frac{V}{w_f(R+y_s)^2 Am} W_f \left(\frac{h^2}{8} - \frac{y_s^2}{2} \right) \quad (2.85)$$

To be able to describe the shear stress in the web, the integration need to be divided into two parts since the active width are different over the region. Equation (2.87) gives shear stress in the web.

$$\tau_{web} = \frac{V}{w_{web}(R+y_s)^2 Am} \left(\int_{y_s}^{h_{web}/2} w_{web} y \, dy + \int_{h_{web}/2}^{h/2} w_f y \, dy \right) = \quad (2.86)$$

$$= \frac{V}{w_{web}(R+y_s)^2 Am} \left(w_{web} \left(\frac{h_{web}^2}{8} - \frac{y_s^2}{2} \right) + w_f \left(\frac{h^2}{8} - \frac{h_{web}^2}{8} \right) \right) \quad (2.87)$$

And finally, equation (2.88) can be used to calculate stress in the lower flange. It shows that the equation for shear stress in the lower flange is exactly the same as for the upper flange.

$$\tau_{lf} = \frac{V}{w_f(R+y_s)^2 Am} \left(\int_{y_s}^{-h_{web}/2} w_f y \, dy + \int_{-h_{web}/2}^{h_{web}/2} w_{web} y \, dy + \int_{h_{web}/2}^{h/2} w_f y \, dy \right) \quad (2.88)$$

Since the integral over the web is symmetrical, it will become zero. Thus

$$\tau_{lf} = \frac{V}{w_f(R+y_s)^2 Am} \left(w_f \left(\frac{(-h_{web})^2}{8} - \frac{y_s^2}{2} \right) + w_f \left(\frac{h^2}{8} - \frac{h_{web}^2}{8} \right) \right) \quad (2.89)$$

Some terms can be terminated, and then the final expression valid for the lower flange (2.90) is obtained:

$$\tau_{lf} = \frac{V}{w_f(R+y_s)^2 Am} \left(w_f \left(\frac{h^2}{8} - \frac{y_s^2}{2} \right) \right) \quad (2.90)$$

2.4.6 STRESS PERPENDICULAR TO LINE OF GRAVITY

To derive an expression for stress perpendicular to center of gravity, vertical equilibrium is studied. As for the derivation of τ , the influence of local external load is omitted.

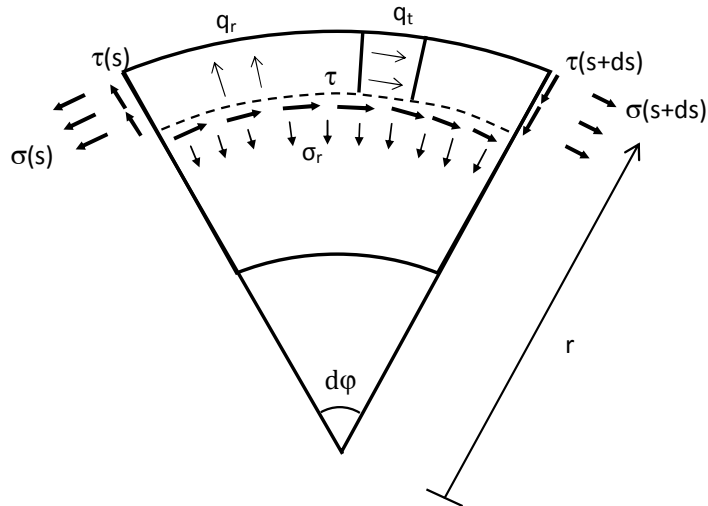


Figure 2.18. Derivation of stress perpendicular to line of gravity.

The same variables as in chapter 2.4.5 are used, except b here denotes width. Vertical equilibrium gives the general expression of stress perpendicular to line of gravity:

$$\int_{As} \tau(s) dA - \int_{As} \tau(s+ds) dA - d\varphi \int_{As} \sigma(s+ds) dA - \sigma_r b r d\varphi = 0 \quad (2.91)$$

Equation (2.32) is inserted

$$\int_{As} \tau(s) - \tau(s+ds) dA - \frac{ds}{R} \int_{As} \sigma(s+ds) dA - \sigma_r b r \frac{ds}{R} = 0 \quad (2.92)$$

Equation (2.92) is rearranged

$$\int_{As} \frac{\tau(s) - \tau(s+ds) dA}{ds} - \int_{As} \frac{\sigma}{R} dA - \sigma_r b \frac{r}{R} = 0 \quad (2.93)$$

By using the fundamental theorem of calculus:

$$\int_{A_s} \tau' - \frac{\sigma}{R} dA - \sigma_r b \frac{r}{R} = 0 \quad (2.94)$$

If (2.96) is solved with respect to σ_r :

$$\sigma_r = \frac{R}{b_{ys}(R+y)} \int_{A_s} \tau' - \frac{\sigma}{R} dA \quad (2.95)$$

The following are equations are known from (2.59), (2.70) and (2.80).

$$\sigma = \frac{N}{A} - \frac{M}{AR} \left(1 + \frac{y}{m(R+y)} \right) \quad (2.96)$$

$$\frac{dV}{ds} = -\frac{N}{R} \quad (2.97)$$

$$\tau = \frac{V}{w_{ys}(R+y_{ys})^2 A m} \int_{A_s} y dA \quad (2.98)$$

It is easy to realize that (2.95) fully expanded using the equations above become very complex and ungainly. By omitting the influence of the normal force and disregarding the curvature effect on the normal stress distribution as suggested by Young (1989) equation (2.95) takes the form (2.100). If effect of curvature is disregarded from, stress can be calculated using equation (2.96). See e.g. Heyden (2005).

$$\sigma = \frac{-My}{I} \quad (2.99)$$

Where M denotes moment, I denote moment of inertia and y is a variable. This simplification can be done since these two factors are very small in comparison to the moment unless the curvature is extremely steep, which seldom is the case for beams used as structural base.

$$\sigma_r = \frac{M}{w_y(R+y_{ys})^2 A m} \int_{A_s} y dy \quad (2.100)$$

As with the shear force, the integral inside (2.100) needs to be divided into three parts depending on where the stress is to be evaluated.

Using (2.81) with (2.100), the top flange stress can be calculated:

$$\sigma_{r,tf} = \frac{M}{w_f(R+y_s)I} \int_{y_s}^{h/2} y \, dy \quad (2.101)$$

Evaluation gives:

$$\sigma_{r,tf} = \frac{M}{w_f(R+y_s)I} W_f \left(\frac{h^2}{8} - \frac{y_s^2}{2} \right) \quad (2.102)$$

For evaluating the web, the integral needs for obvious reasons be divided into two parts:

$$\sigma_{r,web} = \frac{M}{w_{web}(R+y_s)I} \left[\int_{y_s}^{h_{web}/2} y \, dy + \int_{h_{web}/2}^{h/2} y \, dy \right] \quad (2.103)$$

If the integral is evaluated, a final expression is derived:

$$\sigma_{r,web} = \frac{M}{w_{web}(R+y_s)I} \left(W_{web} \left(\frac{h_{web}^2}{8} - \frac{y_s^2}{2} \right) + W_f \left(\frac{h^2}{8} - \frac{h_{web}^2}{8} \right) \right) \quad (2.104)$$

For evaluated the lower flange, integration needs to be carried out over all three sub regions.

$$\sigma_{r,lf} = \frac{M}{w_l(R+y_s)I} \left[\int_{y_s}^{-h_{web}/2} y \, dy + \int_{-h_{web}/2}^{h_{web}/2} y \, dy + \int_{h_{web}/2}^{h/2} y \, dy \right] \quad (2.105)$$

The integral over the web will become zero and can be omitted:

$$\sigma_{r,lf} = \frac{M}{w_l(R+y_s)I} \left[W_f \left(\frac{(-h_{web})^2}{8} - \frac{y_s^2}{2} \right) + W_f \left(\frac{h^2}{8} - \frac{h_{web}^2}{8} \right) \right] \quad (2.106)$$

If (2.106) is rearranged, a final expression for the stress perpendicular to line of gravity for the lower flange is obtained:

$$\sigma_{r,lf} = \frac{M}{w_l(R+y_s)I} \left(W_f \left(\frac{h^2}{8} - \frac{y_s^2}{2} \right) \right) \quad (2.107)$$

2.5 EFFECTIVE STRESS AT COMBINED STATES OF STRESS

In the case of combined states of stress with two or more non-zero stress components, a scalar effective stress can be calculated for evaluation of failure. The different components (normal tangential stress, shear stress and normal stress perpendicular to grain) interact which makes it insufficient to study them as separated units. Empirical expressions are often used. Norris (1962) suggests the following failure criterion:

$$\sqrt{\left[\left(\frac{\sigma}{f}\right)^2 + \left(\frac{\sigma_r}{f_r}\right)^2 + \left(\frac{\tau}{f_v}\right)^2\right]} \leq 1 \quad (2.108)$$

This criterion is partly derived from a theoretical base and is commonly used. σ , σ_r and τ are the stresses in the wood. f , f_r and f_v are the corresponding ultimate stresses when only one stress component is non-zero.

f , f_r and f_v have different values. If the quotient for each of the stresses σ , σ_r and τ over f , f_r and f_v respectively are calculated and then multiplied with stress perpendicular to grain, an effective stress corresponding to the above Norris failure criterion is obtained:

$$\sigma_{r,e} = \sqrt{\left[\left(\frac{\sigma}{f/f_r}\right)^2 + (\sigma_r)^2 + \left(\frac{\tau}{f_v/f_r}\right)^2\right]} \quad (2.109)$$

f , f_r and f_v can be found in design code literature, see e.g. Isaksson and Mårtensson (2006).

2.6 SIZE EFFECT AND WEIBULL THEORY

In brittle materials such as wood, the size has influence on the strength. This is explained by statistics; when a larger volume is under stress the probability that a small portion of the beam has a weakening defect such as a twig becomes more likely. A theory that describes this behavior was developed by Weibull and is referred to as Weibull theory or the weakest link theory (Vännman, 2002). Hult (1966) recites the Weibull cumulative distribution function (2.110).

$$S = 1 - e^{-(\sigma/\sigma_0)^m} \quad (2.110)$$

S is the probability that a test body of unit volume exposed to uniform tension will fail at a tensile tension stress less or equal to σ_B . σ_0 is chosen so that the failure stress is less than σ_0 in 63.2% of the test cases. σ_0 is a material parameter and can also be chosen as the median ultimate stress level of a test body with a volume of $\ln 2$ (≈ 0.69). m^* is the second material parameter and is a measure of the spread. Gustafsson (1983) described a method of application of Weibull theory in relation to strength analysis of concrete pipes. This method is applicable to brittle materials in general, including wood, and is the method used in the present application.

The Weibull distribution describes the probability that a specific piece of the material has a certain ultimate stress level. This means that different parts of a larger piece of the material can withstand different maximum stress. As a result, a larger volume is more likely to have a lower ultimate stress value *somewhere* than a small volume. As for brittle materials, a collapse of one point anywhere in the volume is assumed to lead to total collapse.

The probability gained by using equation (2.110) is valid for a unit volume under evenly distributed stress. In order to use the probability function for analysis of an arbitrary (large) volume with arbitrary stress distribution, the (large) volume can be treated as a number of small unit volumes under evenly distributed stress. The probability that the large volume collapses is equivalent to the probability that a small unit volume collapses somewhere in the large volume. This gives, see (Gustafsson, 1983)

$$S = 1 - e^{-\int_{V_b} (\sigma_i / \sigma_0)^{m^*} dv} \quad (2.111)$$

where V_b is the volume of the structure examined. Using the method derived in this thesis, an analytic expression describing the combined stresses in an arbitrary volume is probably possible to derive but most likely not very useful since of its complexity. A better approach is to numerically calculate the integral stated in equation (2.111) above. Discretization of equation (2.111) yields:

$$S = 1 - e^{-\sum_{i=1}^n (\sigma_{0i} / \sigma_0)^{m^*} \cdot \Delta v_i} \quad (2.112)$$

Using equation (2.109), it is possible to calculate an effective stress in an arbitrary point of a curved beam element. The stress is then summed over the

entire body using equation (2.112) above. The result of such a calculation is the probability (S) that the entire body will collapse.

However, calculation of the probability of failure are seldom made by practicing engineers when designing structures. Instead, prediction of a failure load or a quotient between the actual load and the failure load is preferred. Gustafsson (1983) gives the following equation:

$$\sigma_u = f_t \cdot \left(\frac{V_{prisma}}{V} \right)^{1/m^*} \cdot \gamma^{s^*/m^*} \quad (2.113)$$

where σ_u is the stress in an arbitrary choose reference point when the load equals the median value of the ultimate load. f_t is the median ultimate load for a test body of volume V_{prisma} . m^* is a measurement of the spread for such a test and can be approximately calculated by equation (2.115). V is the volume of the mass under stress and γ is a value defined by:

$$\gamma^{s^*} = \frac{V \cdot \sigma_k^{m^*}}{\int_{V_b} \sigma^{m^*} dV} \quad (2.114)$$

where σ_k is the stress in the reference point as used for equation 2.113 and σ is the stress integrated over the volume. Gustafsson (1983) also shows that the variable m^* can approximately be calculated by

$$m^* \approx - \frac{1.4}{\ln(1-s^*)} \quad (2.115)$$

where s^* is the coefficient of variation.

If (2.114) is inserted into (2.113) it is possible to calculate the quotient between actual stress and ultimate stress in the reference point:

$$\frac{\sigma_k}{\sigma_u} = 1 / \left[f_t \cdot \left(\frac{V_{prisma}}{\int_{V_b} \sigma^{m^*} dV} \right)^{1/m^*} \right] \quad (2.116)$$

The quotient between the stresses is denoted with Q :

$$Q = \frac{\sigma_k}{\sigma_u} \quad (2.117)$$

The expression above will give the same type of result as many formulas used for designing structures; if the quotient is ≥ 1 then the ultimate stress is

reached and the structure will collapse (or to be more exact, the maximum *allowed probability* that the structure will collapse has been reached). The expression is of course dependent on the input parameters f_b , m^* and V_{prisma} . Characteristic value used for design is often set to the 5 percentile. Values for these variables can be derived by testing specimens and can be found tabulated. Thelandersson and Larsen (2003) have tested perpendicular to grain tensile strength of spruce for specimens of various volumes. The following numbers were obtained by testing a prismatic glulam specimen (prisma) of volume $90 \times 275 \times 500 \text{ mm}^3$.

$$\sigma_{5\%} = 0.74 \text{ MPa}$$

$$V_{prisma} = 10 \text{ dm}^3$$

$$m = 5$$

The integral in equation (2.116) can be numerically solved using the same method as used in equation (2.112) and finally a design value for the entire structure can be derived.

CHAPTER 3

3 SOFTWARE IMPLEMENTATION OF THEORY

Many of the calculations presented in the theory chapter are excessively time consuming to do by hand. Since designing structures in many cases is an iterative process it stands clear that the theory needs to be implemented to become truly useful. As a part of this thesis a piece of software has been developed with the aim to completely implement the theory.

The software is developed as a set of MATLAB/CALFEM functions. The toolkit consists of a number of functions which can be used to build up structures made of curved beam elements with hollow box section. The functions for creating and analyzing bars and beams included with CALFEM can be combined with the curved beam functions to analyze complex structures built up by arbitrary FEM-elements.

3.1 OUTPUT

The section forces, the deformed shape and a plot of where the stress is evaluated when using Weibull theory are presented in way that gives the user valuable feedback.

3.2 MATLAB FUNCTIONS

The functions can be found in appendix A. They are commented to make it easier to expand or modify them. The different functions are moreover briefly explained in the following sub chapters.

3.2.1 CBEAM2E

function [Ke,fe] = cbeam2e(ex, ey, eq, ep, flip_beam);

The “curved beam 2 dimension element” uses the method derived in chapter 2.1 to compute the stiffness matrix of a curved beam. First, the modified moment of inertia is calculated using equation (2.9). Then equation (2.4) - (2.7) are used to compute the stiffness matrix. Unless the flag *flip_beam* is set, the beam is “flipped” as described in chapter 2.1.4. Then the stiffness matrix rotated into global position according to method described in chapter 2.1.3.

A basic transformation of distributed load is also implemented. The load is simply divided equally between the two nodes as described in chapter 2.2.2. This is a sufficient approximation if the beam element is very short compared to the size of the arch, and the output force vector should only be used under such conditions. A more refined method for generating the load vector is implemented in the *cbeam2dl* function.

3.2.2 CBEAM2DL

function [Ke,fe] = cbeam2dl(ex, ey, eq, ep, flip_beam, n)

The function “curved beam 2 dimension distributed load” calculates the corresponding load vector of a distributed load. It also returns the stiffness matrix of a curved beam. The function implements the method described in chapter 2.2.1. The curved beam element is divided into n pieces and the distributed load is approximated as $n+1$ point loads. This makes it possible to create a subsystem of smaller curved beam elements where the load attacks directly into the nodal points which make the use of shape functions superfluous. The end points of the beam are then locked and the equation system of the subsystem is solved. The negative reaction force vector is then returned as the corresponding force vector. For convenient reasons, the stiffness matrix is also returned.

3.2.3 CBEAM2DISP

function [ce_disp, mfact] = cbeam2pdisp(ex, ey, eq, ep, flip_beam, n, ed, mfact)

The “curved beam 2 dimension displacement” is used to plot the deformation of a curved beam under load/displacement. The methodology is much the

same as for cbeam2dl. The element is divided into smaller parts, the distributed loads are approximated as point loads and subsystem is then solved using the global displacements as element end boundary conditions.

3.2.4 CBEAM2P

function [beam_coord,xc,yc] = cbeam2p(ex, ey, R, n, flip_beam)

This function uses trigonometry, geometry and algebra to calculate coordinates evenly (angle-wise) spread over a bow of constant radius. The function will return a vector of size [nx2] containing the coordinates of the bow. It will also return the coordinates of the center of the circle the bow is a part of.

3.2.5 CBEAM2STRESS

function [sigma,tau,sigma_perpendicular] = cbeam2stress(ys, M, V, N, ep)

The “curved beam 2 dimension stress”-function calculates the normal stress, shear stress and stress perpendicular to line of gravity at any height of the cross section of a curved beam element made out of a hollow box section. The stresses are computed using the stress equations derived in chapter 2.4.

3.2.6 CBEAM2S

function [es,sf] = cbeam2s(ex, ey, eq, R, n, Q, flip_beam)

The function “curved beam 2 dimension section forces” is used to calculate the section forces of a curved beam element. The section forces are computed using equilibrium conditions as derived in chapter 2.2.4. This also means that this method has the limitation that the section forces (or reaction forces) must be known in the left end of the beam for this function to work.

3.2.7 CBEAM2WEIBULL

function [sum_sigma, ft, m, Vprisma] = cbeam2weibull(ex, ey, es, ep, flip_beam, settings)

The “curved beam 2 dimension weibull”-function is used to analyze the combined stresses using weibull statistics as derived in chapter 2.6. The curved beam element is divided into a number of smaller regions. The number of tangential evaluation points are determined by the number of elements in the es vector, which are individually examined using the cbeam2stress

function. Unless the settings-vector is defined, standard values of V_{prisma} , f_t , m^* , f , f_r and f_v are used. The settings-vector is defined as follows:

$$settings = [m^* f_t V_{prisma} f_v f f_r]$$

The number of evaluation points perpendicular to line of gravity is defined by the variables $iterations_flange$ and $iterations_web$ in the first section of the code. The function returns the integral over V_b , defined by equation (2.118) as well as f_t and m^* . This makes it possible to calculate the collapse probability for structures made up out of several curved beam elements. The function will also plot the points of evaluation, see figure 3.1 below. The figure is a close up of the result of the example in appendix B.

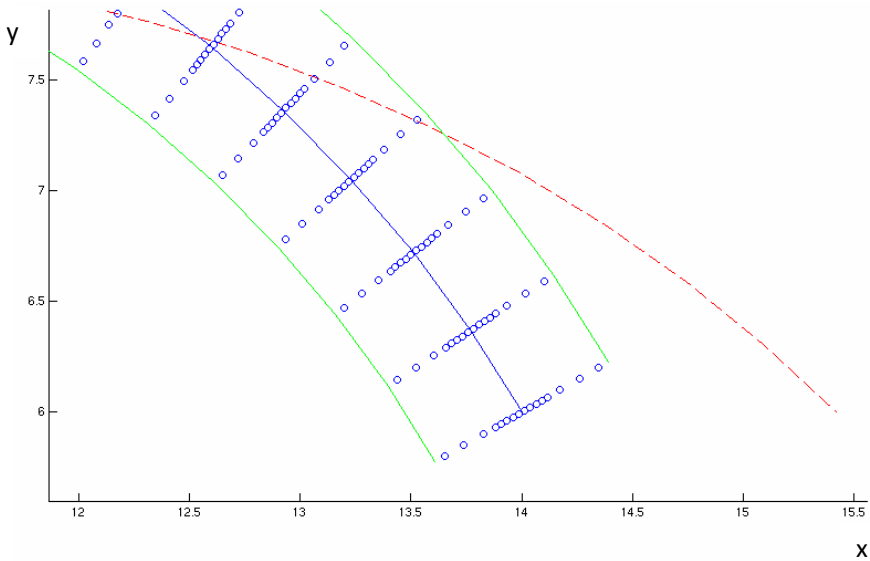


Figure 3.1. Points of evaluation and deformed curvature

CHAPTER 4

4 CURVATURE EFFECT ON STRESSES

The stress distribution in a beam is effected by the curvature. To make the reader attentive on this fact, and to illustrate how the results of the stress equations presented in this paper differs from the distribution in a straight beam, some plots are presented below. The stresses are plotted using the *cbeam2stress* function found in appendix A. The section which has been evaluated is presented in figure 4.1 below. The input load parameters are $N = -10.6 \text{ kN}$, $M = 16.5 \text{ kNm}$, $V = -15.9 \text{ kN}$.

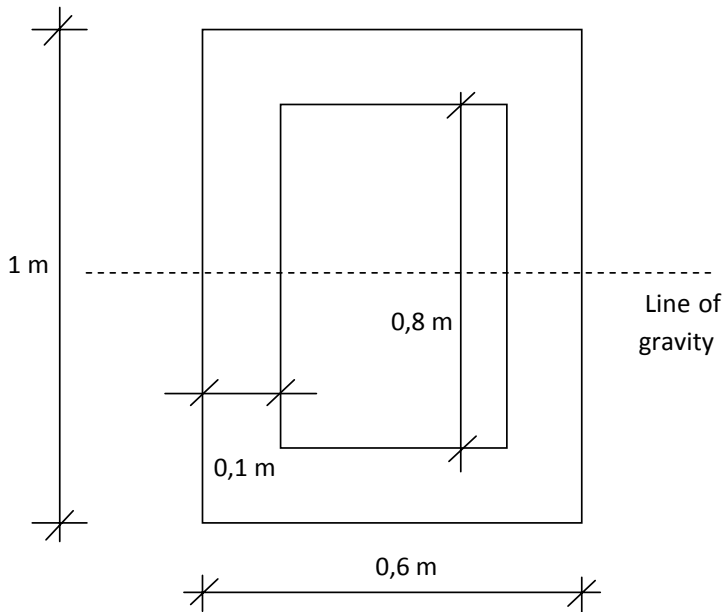


Figure 4.1. Geometry of example beam.

4.1 NORMAL STRESS

In figure 4.2 below, two distributions are plotted. The solid line is valid for a ratio between radius and height of $R/h = 2$ and the dotted line is valid for $R/h = 1000$. All other conditions are the same for the two beams. As seen in the plot, the large radius beam has an almost linear distribution of normal stress. It can also easily be seen that the maximum stress level is underestimated if using a straight-beam formula for a steep curvature beam.

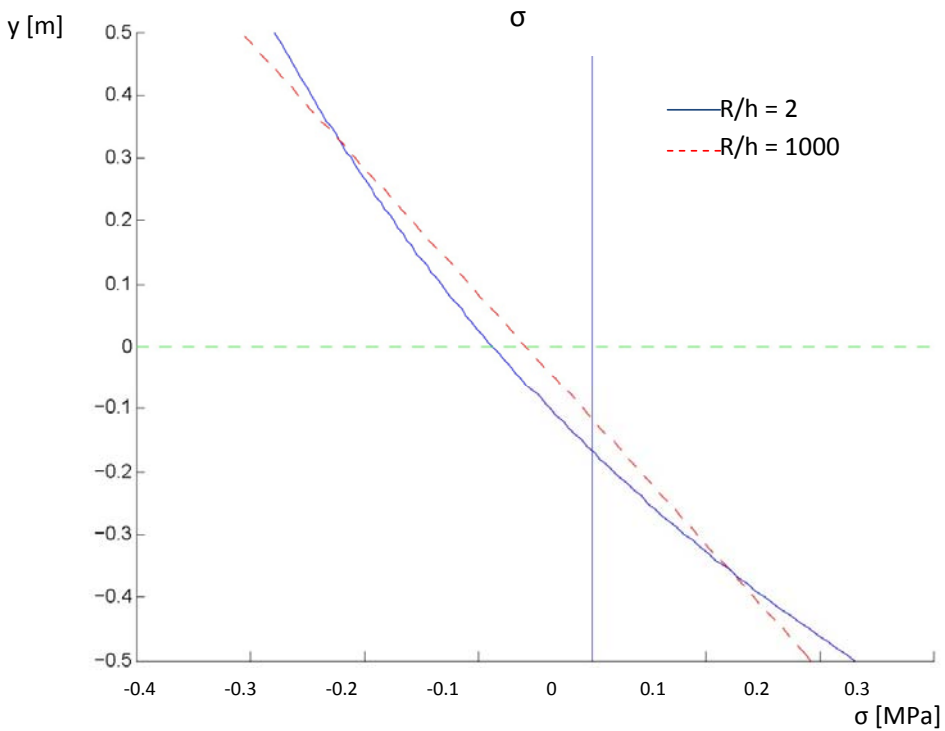


Figure 4.2. Normal stress is dependent on curvature.

4.2 SHEAR STRESS

As for normal stress, shear stress is affected by curvature. The solid line represents a curved beam where $R/h = 2$. The dotted line represents $R/h = 1000$ for the same variables for a beam under the same load. As for the case with normal stress, maximum stress is underestimated if using straight beam formulas for a steep curvature beam. It can also be observed that the stress level of the web is much higher than the stress level for the flange. This is due to the fact that the web has a smaller width than the flange. See figure 4.3 below:

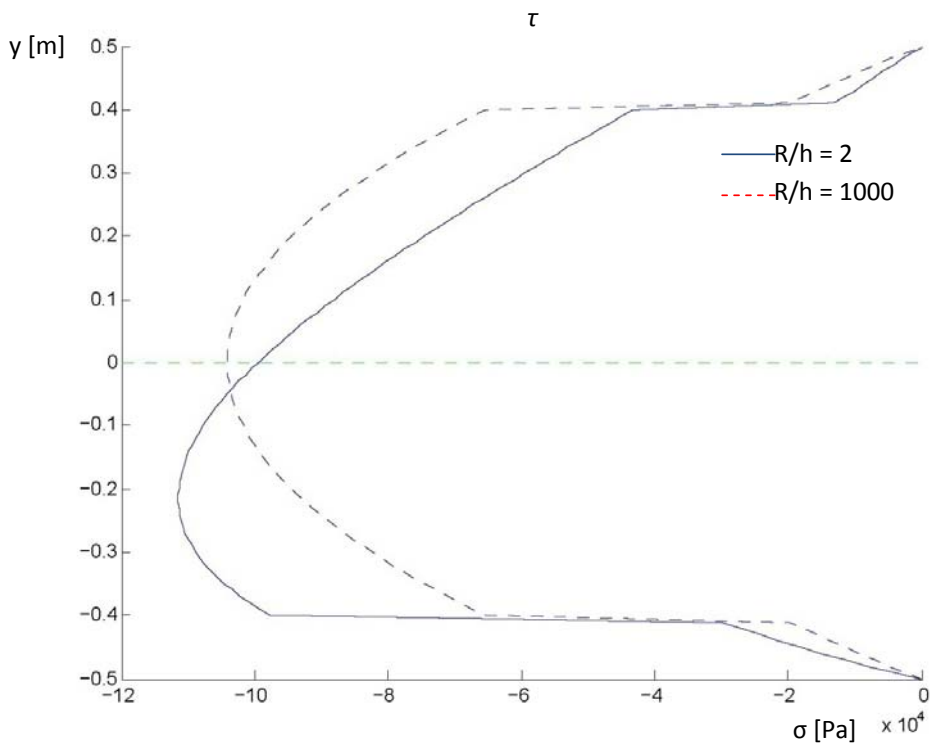


Figure 4.3. Shear stress is dependent on curvature.

4.3 STRESS PERPENDICULAR TO LINE OF GRAVITY

The most interesting effect of introducing curvature on a beam is the appearance of stress perpendicular to line of gravity. This kind of stress does not exist at all in a perfectly straight beam. For steep curvature, this kind of stress becomes quite large and has to be taken into account. This is especially true for wooden beams since the ultimate stress levels perpendicular to line of gravity is just a few percent of the ultimate tension stress levels of normal stress (Burström, 2001). On the plot below, the solid line represents a beam with $R/h = 2$ and the dotted line represents $R/h = 1000$ for the same variables. All other conditions are the same. As seen in the figure, the presence of stress perpendicular to line of gravity is virtually zero for an almost straight beam. For beams of very steep curvature, it will in most cases become the limiting factor for the load capacity. See figure 4.4 below:

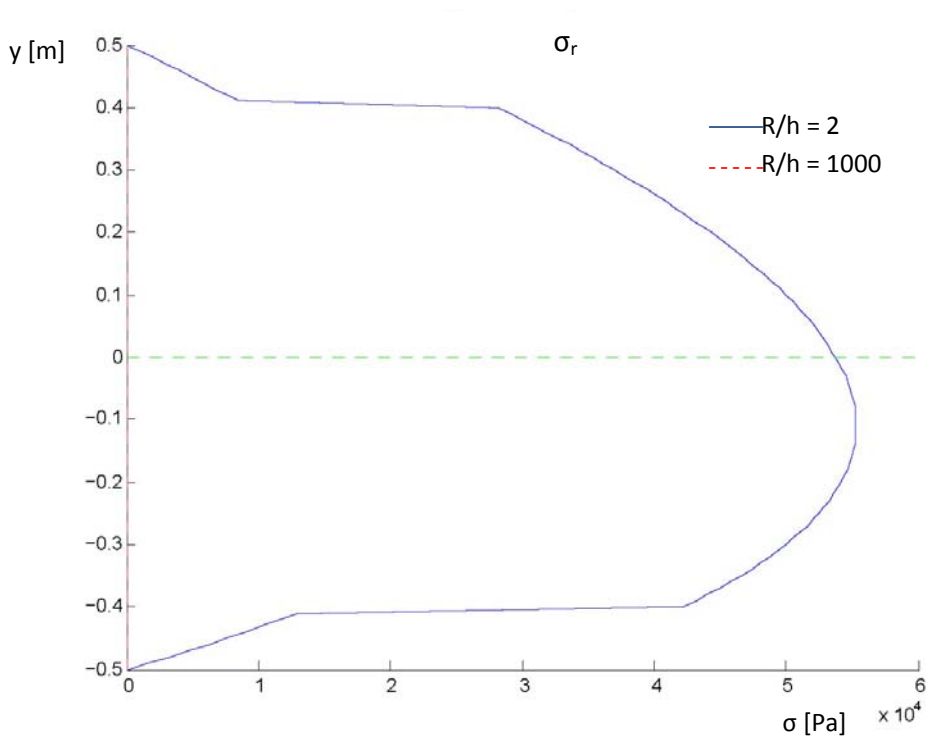


Figure 4.4. Stress perpendicular to line of gravity is dependent on curvature.

4.4 COMMENTS ON CURVATURE EFFECT ON STRESSES

It is noted that the absolute maximum of stress perpendicular to line of gravity and shear stress does not appear to occur at the same level as the neutral axis as expected. This may however depend on the fact that some simplifications have been made during derivation of the stress formulas. The equation used for stress perpendicular to line of gravity disregards the effect curvature has on normal stress distribution. The formula relies on a linear variation of normal stress (dotted line in figure 4.2) rather than the more exact curved distribution (solid line in figure 4.2). The influence of the simplifications made become evident for beams with small R/h . However, the offset decreases as the curvature decreases which makes the equations sufficient as long as the curvature is not extremely steep.

CHAPTER 5

5 USER MANUAL – CBEAM2 TOOLKIT EXPLAINED

This chapter describes how to use the cbeam2 toolkit to analyze the strength of a structure built up by curved beam elements. The software required is MATLAB, a numerical computing environment and the programming language, CALFEM, a finite element toolbox developed as an extension to MATLAB by the Division of Structural Mechanics, Lunds University, and the set of CBEAM2-functions listed in appendix A. The arch structure in figure 5.1 below is to be analyzed.

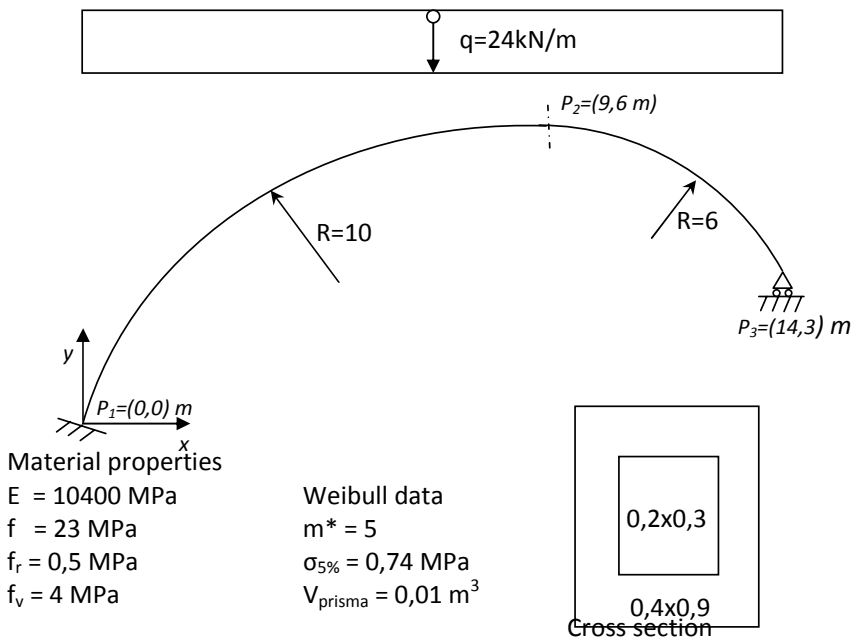


Figure 5.1. Arch geometry and properties

The user must first make sure that the location of both the CALFEM-toolbox and the CBEAM2-package is listed in the environmental variable PATH of MATLAB. To add a directory to the PATH, choose *<Set Path>* under the menu *<File>* of MATLAB. The CALFEM-toolkit can be found on the website of Department of Structural Mechanics at the University of Lund (which as of 2008-05-14 is <http://www.gorkon.byggmek.lth.se/bmforms/AppListPage>). The CBEAM2-package can be received by contacting the author of this thesis. Lines starting with the sign '*>*' states what is to be written into the MATLAB prompt.

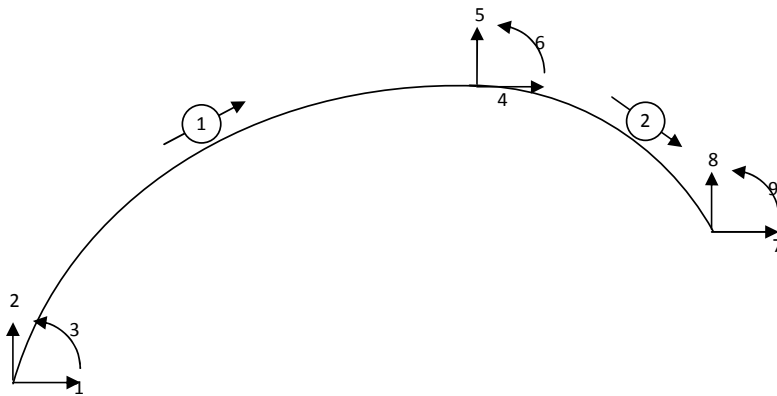


Figure 5.2. Element and DOF numbering

First, the geometry needs to be defined for the two beam elements that the arch consists of. The variables e_x and e_y holds information about x and y coordinates. The e_p -vector holds information about Young's modulus, radius, total width of cross section, total thickness of walls, total height and height of web:

```
> ex1 = [0 9];
> ey1 = [0 6];
> ep1 = [10400e6 10 0.4 .2 .9 .3];
> ex2 = [9 14];
> ey2 = [6 3];
> ep2 = [10400e6 6 0.4 .2 .9 .3];
```

Then the load needs to be defined. The e_q -vector holds information about vertical and horizontal load:

```
> eq1 = [0 -24000];
> eq2 = [0 -24000];
```

The variable `n` denotes the number of evaluation points used for analysis of each arch. In this example 15 points are used for both beam elements.

```
> n = 15
```

When load and geometry is defined, it is possible to create the stiffness matrix and load matrix using the `cbeam2d1()` function.

```
> [Ke1, fe1] = cbeam2d1(ex1, ey1, eq1, ep1, 0, n)
> [Ke2, fe2] = cbeam2d1(ex2, ey2, eq2, ep2, 0, n)
```

The element degrees of freedom are defined by the `edof`-matrix, see figure 5.2.

```
> edof = [1 1 2 3 4 5 6
          2 4 5 6 7 8 9];
```

The boundary conditions are defined using the `bc`-matrix. Since the left end of the arch is fix-ended and the right end is simply supported the `bc`-matrix becomes:

```
> bc = [1 0
        2 0
        3 0
        8 0];
```

An empty global stiffness matrix and an empty global force matrix needs to be created, preferably by the built in command `zeros()`.

```
> K = zeros(9);
> f = zeros(9,1);
```

The two beam elements needs to be assembled to create a global system. The CALFEM-function `assem()` is used for this purpose. `assem()` makes use of the `edof`-matrix to complete its task.

```
> [K, f] = assem(edof(1,:), K, Ke1, f, fe1)
> [K, f] = assem(edof(2,:), K, Ke2, f, fe2)
```

The system is now complete with global stiffness matrix, global load vector and defined boundary conditions. The CALFEM-function `solveq()` is used to solve the equation system $Ka = f$ (equation 2.2).

```
> [a, Q] = solveq(K, f, bc);
```

The displacements and reaction forces of the global system is now known. The `cbeam2p()` and `plot()` is used to plot the undeformed geometry and `cbeam2pdisp()` can be used to plot the deformed geometry:

```
> figure; hold on;
> axis square;
> axis([0 20 -13 7]);
> title('Analysis of an arch using the cbeam2
> toolkit');

> bcoord1 = cbeam2p(ex1, ey1, ep1(2), n, 0);
> bcoord2 = cbeam2p(ex2, ey2, ep2(2), n, 0);

> plot(bcoord1(:,1), bcoord1(:,2), '-');
> plot(bcoord2(:,1), bcoord2(:,2), '-');

> [ce_disp, mfact] = cbeam2pdisp(ex1, ey1, eq1, ep1,
0, n, a(1:6), 0);
> cbeam2pdisp(ex2, ey2, eq2, ep2, 0, n, a(4:9),
mfact)
```

The `ce_disp` vector contains the geometry of the deformed beam element. The `mfact` is the magnification factor which is calculated by the function unless it is defined in the call. To compute the section forces the command `cbeam2s()` is used. Since the section forces are calculated using equilibrium condition for a subsystem of the beam, the reaction forces of the left end of the beam element needs to be known.

```
> [es1, sf] = cbeam2s(ex1, ey1, eq1, ep1(2), n,
Q(1:3), 0);
> [es2, sf] = cbeam2s(ex2, ey2, eq2, ep2(2), n, sf,
0);
```

The CALFEM function `eldia2()` is used to plot the section forces for the two beam elements.

```
> sfac = eldia2(ex1, [-3 -3], es1(:,3));
```

```

> sfacv = eldia2(ex1, [-6 -6], es1(:,2));
> sfacn = eldia2(ex1, [-9 -9], es1(:,1));
> eldia2(ex2, [-3 -3], es2(:,3), [2 1], sfac);
> eldia2(ex2, [-6 -6], es2(:,2), [2 1], sfacv);
> eldia2(ex2, [-9 -9], es2(:,1), [2 1], sfacn);

```

The function `cbeam2weibull()` makes use of the Weibull theory described in chapter 2.6, `ssigma` is the value of the integral described in equation (2.115):

```

> [S1, ft, m, Vprisma] = cbeam2weibull(ex1, ey1, es1,
ep1, 0, 0);
> [S2, ft, m, Vprisma] = cbeam2weibull(ex2, ey2, es2,
ep2, 0, 0);

```

The function `cbeam2weibull()` does plot a dot for every point of evaluation. When the value of the integral has been evaluated for both beam elements the probability of collapse can be calculated using equation (2.116):

```

> Q = 1/(ft * (Vprisma/(S1+S2))^(1/m))

```

Q =

0.972

Where Q denotes the quotient between maximum allowed value of stress and actual value of stress as described in chapter 2.6. If Q is 1 or lower, the maximum stress has not been reached and the building will not collapse. Of course, if the arch would have consisted of more beam elements, the sum ' $S1+S2$ ' will have to be expanded to include those elements as well. The total set of commands can be found in appendix B. The graphical output is presented in figure 5.3:

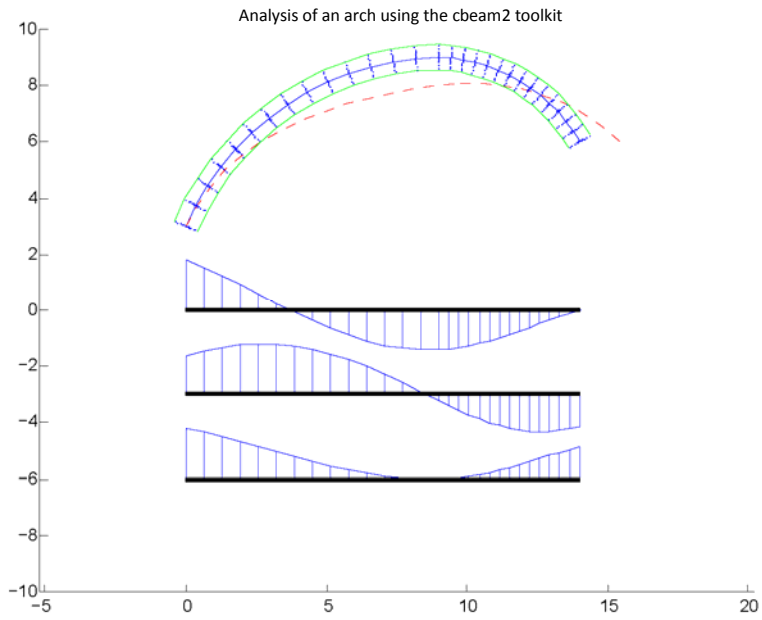


Figure 5.3. Output from example in appendix B.

CHAPTER 6

6 EFFECT OF LOAD DISTRIBUTION, SIZE AND SHAPE ON STRENGTH

When designing an arch, parameters such as height and size of the cross section have an influence on the strength of the structure. In this chapter a number of setups of arch structures are studied using the cbeam2 toolbox.

6.1 INFLUENCE OF LOAD DISTRIBUTION AND SIZE

To study the influence of size and distribution of load on strength, the following setups are analyzed:

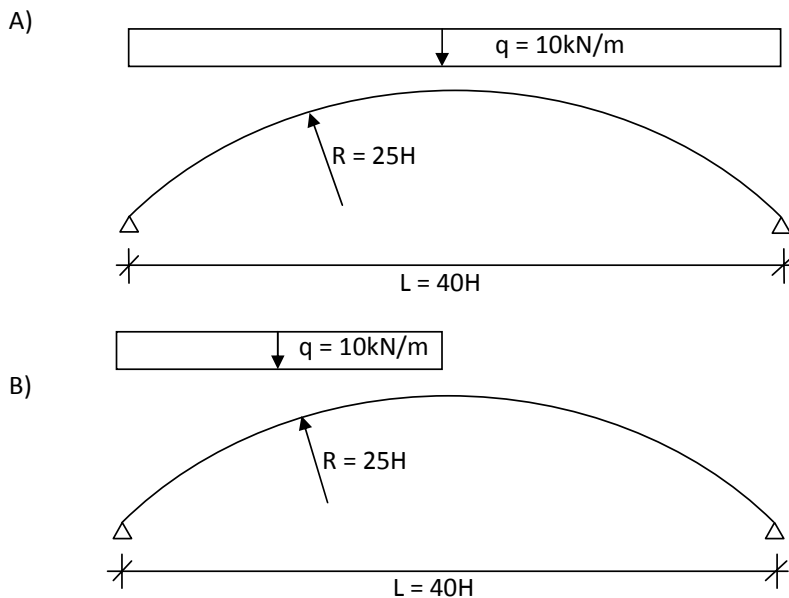


Figure 6.1. Geometry and load for evaluation of size effect

The size of the cross section is shown in figure 6.2:

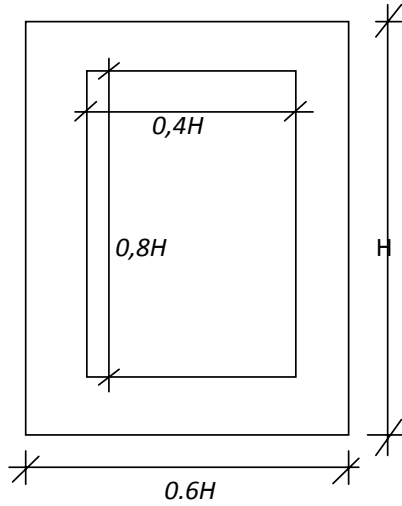


Figure 6.2. Definition of cross section

The result of the analysis is presented in figure 6.3 below. The variable Q on the y-axis is stress over ultimate stress and H on the x-axis is the parameter that defines the size of the structure. Hence; if $Q > 1$ the structure will collapse.

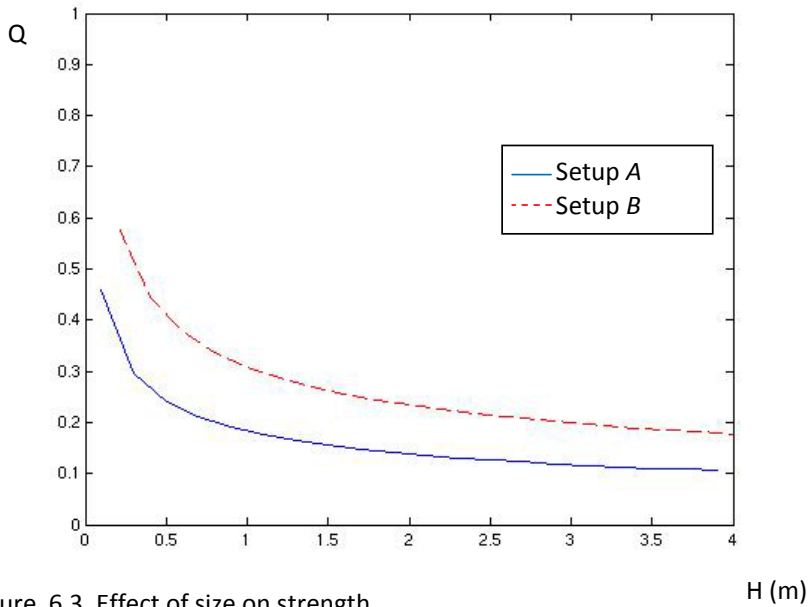


Figure 6.3. Effect of size on strength

The two curves in figure 6.3 shows that the probability of collapse decreases as the structure becomes larger in comparison to the load. This is expected as a larger structure has a larger volume which decreases the average stress. However, the positive effect on strength by scaling up the structure decreases as the volume increases. This is expected in accordance to Weibull theory which also is known as the size effect. Since a larger structure contains a larger volume of wood under stress, it is more likely that a small portion of that volume has a weakening defect. As this probability of a weakening defect increases, the positive effect of a larger structure in comparison to the load decreases.

6.2 INFLUENCE OF SHAPE AND HEIGHT

To study the influence of shape and height on strength, the following setups are analyzed:

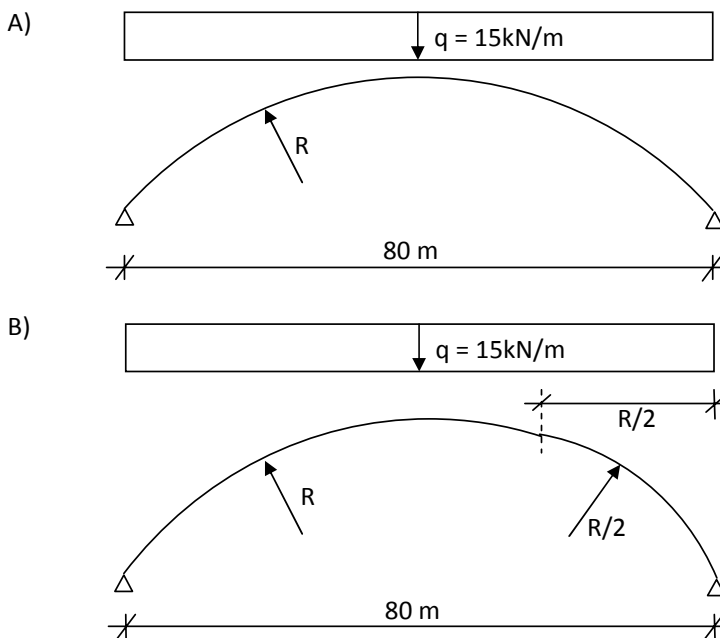


Figure 6.4. Geometry and load for evaluation of shape effect

The size of the cross section is shown in figure 6.5:

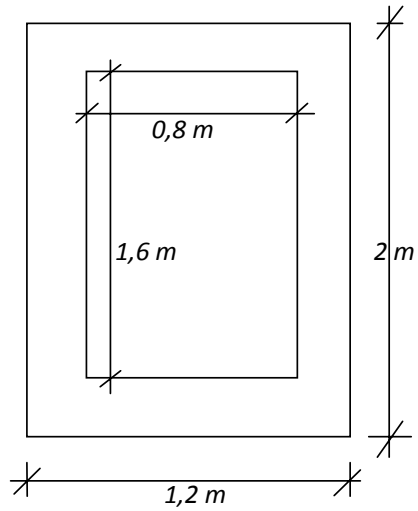


Figure 6.5. Definition of cross section

The result of the analysis is presented below in figure 6.6:

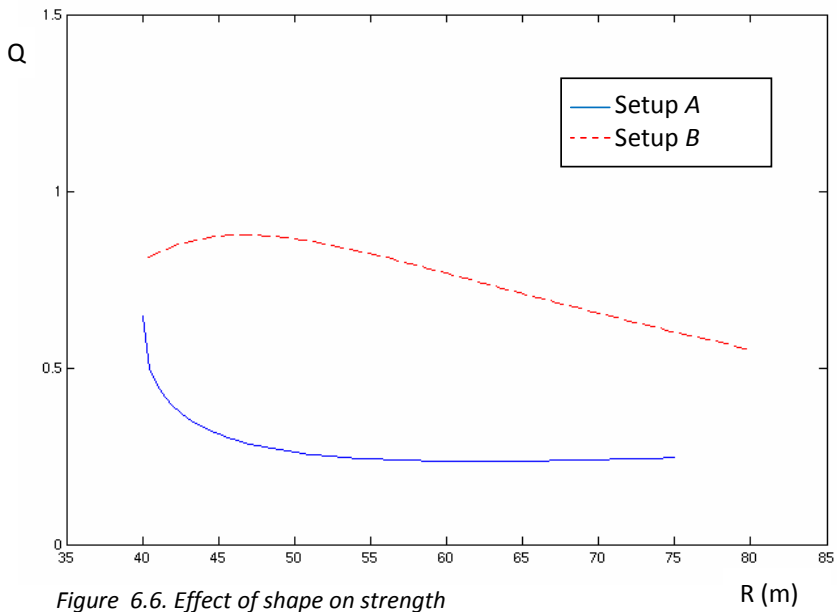
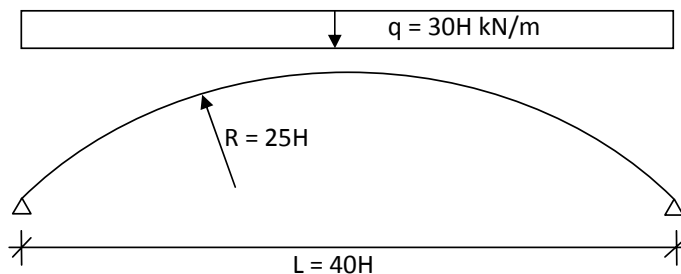


Figure 6.6. Effect of shape on strength

Q is same quotient as for the setups in chapter 6.1. For setup A, the lowest probability of collapse seems to occur when H is about 64. When H is equal to 40, the shape of the arch becomes a half circle. As H increases, the curvature of the arch decreases and as H goes towards infinity, the arch will assume the shape of a straight beam. It is possible to design an arch in such a way that no bending moment occurs for a specific load. Such a design is said to follow the thrust line. For a uniform load the thrust line will become parabolic (Olsson, 2001). With this in mind, it makes sense that the most efficient design for the arch in setup A is somewhere between a half circle and a straight line. Setup B is harder to analyze since the shape of the arch is more complex. The reason may be that as H increases, the arch will get a more efficient curvature. For setup B there will be infinitely high stresses ($R=0$) in the intersection point between the beam elements, these are disregarded of.

6.3 INFLUENCE OF SIZE EFFECT

To analyze the influence of the size effect on strength, the following setup is analyzed:



6.7. Setup for analyze of size effect

The size of the cross section is the same as used in chapter 6.1 and is presented in figure 6.2. By altering the parameter m^* in equation 2.116, which denotes spread in Weibull theory, it is possible to reduce the influence of the size effect on strength. As m^* approaches infinity the standard deviation approaches 0 and hence there will be no variation of the ultimate stress value. The setup in figure 6.7 has been analyzed for $m^*=5$ and $m^*=50$ for different H. The result is presented in figure 6.8:

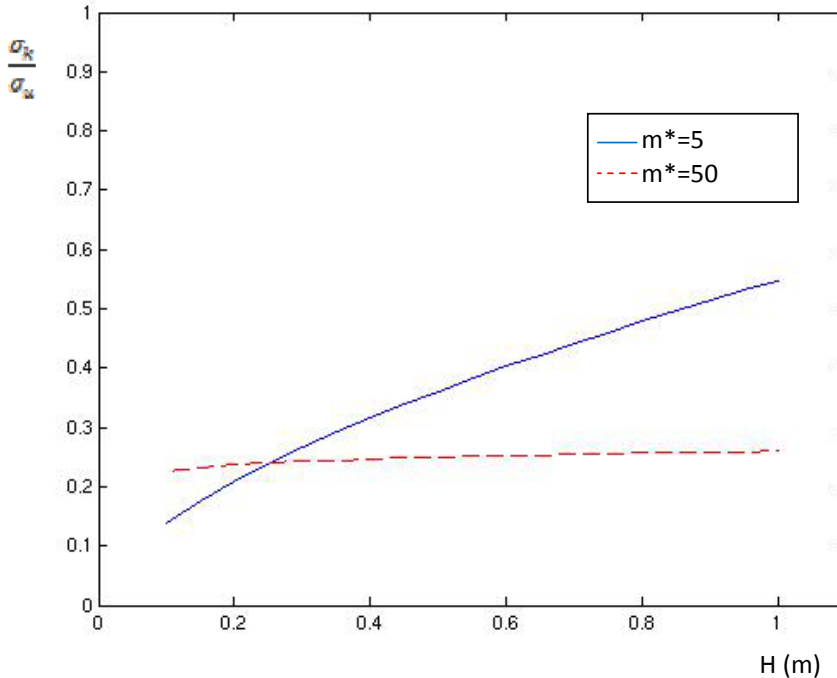


Figure 6.8. Influence of size effect

If m^* would be infinite, the full line in figure 6.8 above would form a straight line. The result shows what is expected; if the size effect is disregarded from there is no effect of scaling the structure up as shown by the (almost) straight line in figure 6.8. If the size effect is taken into account scaling the structure up results in a higher probability of failure.

6.4 ANALYSIS OF ARCH 7 IN SANDVIKEN STRUCTURE

As mentioned in the first chapter, a structure built up of glulam arches of box-section are being built in Sandviken, Sweden, during spring 2008. In this chapter one of the structural arches is analyzed using both a deterministic method and the more refined method based on Weibull theory described in this thesis. As explained in chapter 6.3 the Weibull theory will act as a deterministic method if the spread parameter m^* is infinite. Since m^* acts as a power function, $m^*=50$ is a very large value. The geometry of the arch is defined in figure 6.9. The material properties are the same as in figure 5.1, except m^* .

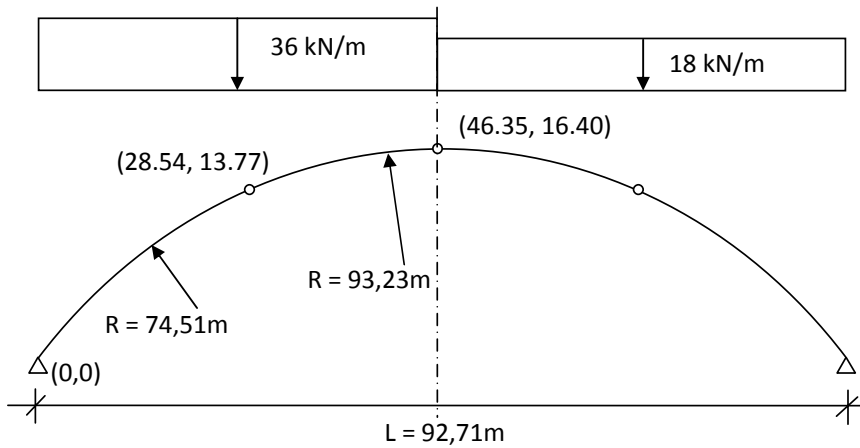


Figure 6.9. Analysis of size effect on strength

The arch consists of four curved beam elements with different radius. The arch is symmetrical and pin-ended. The load case represents an unevenly distributed snow load. The cross-section of the arch is presented in figure 6.10 below:

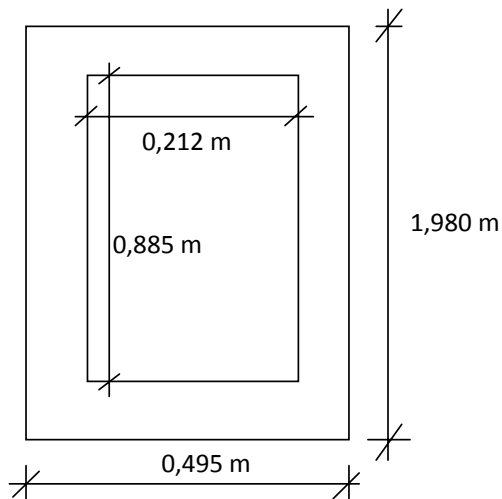


Figure 6.10. Cross section

The results show that the size effect has quite a large effect on structure of such large volume. The output of the CBEAM2-toolbox is presented in figure 6.11 below:

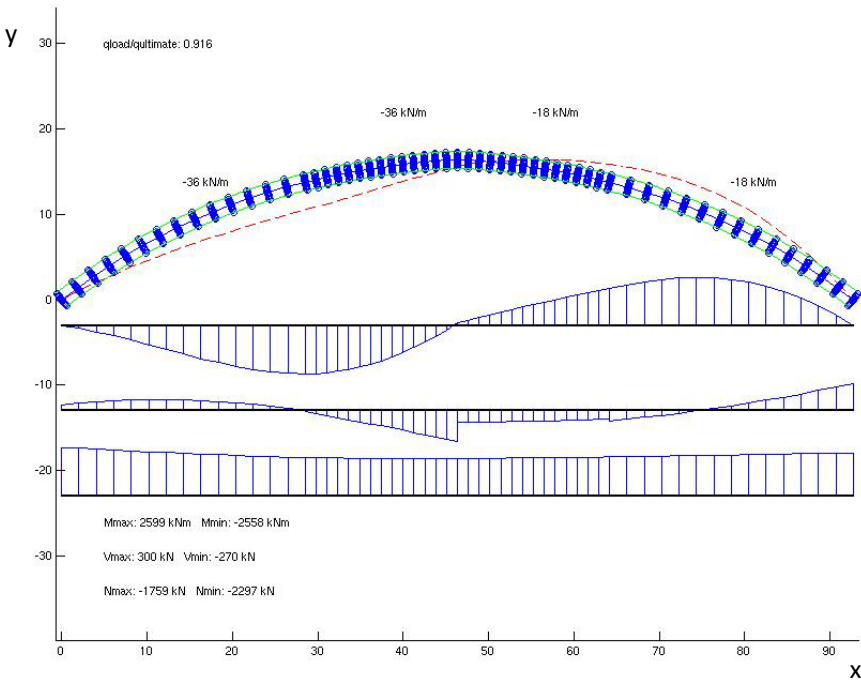


Figure 6.11. Analysis of arch when size effect is disregarded of.

In figure 6.11, the plots of the section forces are not plotted using the same scale. In the plot above (figure 6.11) the Weibull spread parameter m^* is set to 50. This means the size effect is almost completely disregarded of. The quotient between stress and ultimate stress is for different m^* :

$$m^* = 5 \rightarrow \frac{\sigma_K}{\sigma_U} = 0,916$$

$$m^* = 50 \rightarrow \frac{\sigma_K}{\sigma_U} = 0,315$$

Hence, according to Weibull theory the maximum allowed stress is about 1/3 of the maximum allowed stress calculated by deterministic theory for this particular structure under this specific load. It should be noted that values presented above are of characteristic value since the input value of the parameter f_r used is the characteristic value, see figure 5.1 and equation 2.108.

CHAPTER 7

7 CONCLUSIONS

7.1 CONCLUDING REMARKS

This toolbox offers powerful tools for analysis of stress in curved beams. The results show that the Weibull theory size effect has a significant impact on strength for large beams. The result of the toolbox has been validated by comparing the output of the different functions with hand calculations based on straight beam theory. However, before the toolbox can be used in a professional matter, the correctness should be confirmed by comparing results with other analysis.

7.2 FUTURE WORK

This report makes a base for further work on the subject. The following areas have been identified as prosperous for continues work by the author:

- Derive the load vector the proper way using shape functions. Since the shape functions for a curved beam can be derived, even though they become complex, this could be done. This would save much computing time comparing to the method described in this paper.
- Optimize the MATLAB-functions. There are several steps in the functions (especially where the beam is divided into a subsystem) which easily can be optimized. For example, the un-rotated stiffness matrix of a subsystem beam could be reused to save computing time.
- Implement the MATLAB functions in a high-speed optimized language such as FORTRAN.
- Create a graphical interface to let the user design curved beams without having to learn CALFEM/MATLAB.

BIBLIOGRAPHY

- Austrell, P-E., et. al., 2004. *CALFEM – A finite element toolbox version 3.4*. KFS i Lund AB, Lund.
- Burström, P. G., 2001. *Byggnadsmaterial*. Studentlitteratur, Lund, Sweden.
- Bresler, B., Lin, T. Y., 1960. *Design of Steel Structures*. John Wiley, New York.
- Gustafsson, P. J., 1983. *Oarmerade betongrörs böjbrottlast och ringbrottlast – teoretiska beräkningsmetoder*. Avdelningen för Byggnadsmateriallära, Tekniska högskolan i Lund. Rapport TVBM-3012.
- Heyden, S., et. al., 2005. *Introduktion till strukturmekniken*. KFS i Lund AB, Lund, Sweden.
- Hult, J., 1966. *Hållfasthetslära*. Almqvist & Wiksell/Gebbers Förlag AB, Stockholm, Sweden.
- Ingelstam, E., Rönngren, R., Sjöberg, S., 2004. *Handbok för grundläggande teknisk fysik, fysik och matematik*. KFS i Lund AB, Lund, Sweden.
- Isaksson, T., et. al., 2005. *Byggkonstruktion*. Studentlitteratur, Lund, Sweden.
- Isaksson, T., Mårtensson, A., 2006. *Byggkonstruktion tabell- och formelsamling*. Studentlitteratur, Lund, Sweden.
- Krenk, S., 1993. *A general format for curved an non-homogeneous beam elements*. Department of Building Technology and Structural Engineering, University of Aalborg, Aalborg, Denmark.
- Litewka, P., Rakowski, J., 1996. *An efficient curved beam finite element*. Institute of Structural Engineering, Pozna University of Technology, Pozna, Poland.
- Norris, C. B., 1962. *Strength of Orthotropic Materials subjected to Combined Stresses*. Forest Products Laboratory. Report No. 1816.
- Olsson, N., 2001. *Glulam timber arches – strength of splices and reliability-based optimization*. Department of Civil and Mining Engineering, Luleå University of Technology. Doctoral Thesis.

Ottosen, N., Petersson, H., 1992. *Introduction to the Finite Element Method*. Pearson Education Limited, Essex, England.

Thelandersson, S., Larsen, H. J., 2003. *Timber Engineering*. Wiley.

Young, C. W., 1989. *Roark's Formulas for Stress and Strain – Sixth edition*. McGraw-Hill, Inc., USA.

Vännman, K., 2002. *Matematisk statistik*. Studentlitteratur, Lund, Sweden.

Appendix A

Curved beam element functions

```
CBEAM2E(EX, EY, EQ, EP, FLIP_BEAM)
```

```
function [Ke,fe] = cbeam2e(ex, ey, eq, ep, flip_beam);

% function [Ke,fe] = cbeam2e(ex, ey, eq, ep, flip_beam);
%-----
% PURPOSE
%
% Compute the stiffnes matrix of a curved beam element
% made out of a hollow box section.
%
%-----
% INPUT:      ex = [x1 x2]          Coordinates
%             ey = [y1 y2]
%             eq = [qx qy]          Uniform external load (global
%                                 coordinate system)
%             ep = [E              Young's modoulous
%                  R              Radius of curvature
%                  w_flange       Width of flange
%                  w_web          Total width of web ('walls')
%                  htot           Total height of cross section
%                  hweb]          Height of web
%             flip_beam           Boolean, if set to true curvature
%                                 of beam will be flipped. See chapter
%                                 x for more detailed information.
%
%-----
% OUTPUT:     Ke                   Element stiffnes matrix (global)
%             fe                   Element force vector (simple
%                                 approximation, global system)
%
%-----
% AUTHOR:      Erik Persson      2008-04-18
%
% Copyright (c) Erik Persson
%-----

% Element properties
E = ep(1);
R = ep(2);
w_flange = ep(3);
w_web = ep(4);
htot = ep(5);
hweb = ep(6);
```

```

% 2a = Distance from (x1,y1) to (x2,y2)
g=[ ex(2)-ex(1); ey(2)-ey(1) ];
a = sqrt(g'*g)/2;

% Angle fi (see figure x for definition)

fi = asin(a/R);

% Area
A = w_flange*(htot-hweb)+hweb*w_web;

% Modified bending stiffnes for hollow box section
J = (R^3*w_flange*log((2*R+htot)/(2*R-htot))-R^2*w_flange*htot) -
(R^3*(w_flange - w_web)*log((2*R+hweb)/(2*R-hweb))-R^2*(w_flange -
w_web)*hweb);

% Expressions used to build up the D matrix
H11 = 2*( R/(E*A)+R^3/(E*J) ) * fi * (cos(fi)^2) + (R^3/(E*J)) * (fi -
3 * sin(fi) * cos(fi));
H22 = 2*( R/(E*A)+R^3/(E*J) ) * fi * sin(fi)^2;
H12 = 2*( R/(E*A)+R^3/(E*J) ) * fi * sin(fi)*cos(fi) - 2*( R^3/(E*J) )
* sin(fi)^2;
H33 = R^3/(E*J)*(fi-sin(fi)*cos(fi));
H = H11*H22 - H12^2;

% G matrix
G = [-1, 0, 0, 1, 0, 0;
      0, 0,-a, 0, 0, a;
      0, 1, a, 0,-1, a];

% D matrix
D = [H22/H, -H12/H, 0;
     -H12/H, H11/H, 0;
      0, 0, 1/H33];

% Ke in local coordinate system
Ke = G'*D*G;

% Transformation to global system
alpha = asin((ey(2)-ey(1))/(2*a));

b=[ ex(2)-ex(1); ey(2)-ey(1) ];
L=sqrt(b'*b); n=b/L;

Gt=[n(1) n(2) 0 0 0 0;
     -n(2) n(1) 0 0 0 0;
      0 0 1 0 0 0;
      0 0 0 n(1) n(2) 0;
      0 0 0 -n(2) n(1) 0;
      0 0 0 0 0 1];

% Approximation of distributed load
fy = abs((ex(2)-ex(1))/2)*eq(2);
fx = abs((ey(2)-ey(1))/2)*eq(1);

fe = [fx fy 0 fx fy 0]';

% "Flip" beam, unless flip_beam flag is set
if (flip_beam == 0)
    Ke = [ Ke(1,1) Ke(1,2) -Ke(1,3) Ke(1,4) Ke(1,5) -Ke(1,6)

```

```
Ke(2,:)
-Ke(3,1) Ke(3,2) Ke(3,3) -Ke(3,4) Ke(3,5) Ke(3,6)
Ke(4,1) Ke(4,2) -Ke(4,3) Ke(4,4) Ke(4,5) -Ke(4,6)
Ke(5,:)
-Ke(6,1) Ke(6,2) Ke(6,3) -Ke(6,4) Ke(6,5) Ke(6,6)];
end;

% Transform to global system
Ke = Gt'*Ke*Gt;

%-----end-----
```

CBEAM2DL(EX, EY, EQ, EP, FLIP_BEAM, N)

```

function [Ke,fe] = cbeam2dl(ex, ey, eq, ep, flip_beam, n)

% function [Ke,fe] = cbeam2dl(ex, ey, eq, ep, flip_beam, n)
%-----
% PURPOSE
%
% Compute the corresponding force vector of a curved beam element
% under distributed load applied along the axis of the global
% coordinate system. The global load is approximated as a
% number (n) point loads. A subsystem of (n-1) curved beam
% elements is created with fully supported ends. The equation
% system is then solved and the negative supporting forces are
% returned as the corresponding load vector.
%
% For convenient reasons, the stiffness matrix is also returned.
%
%-----
% INPUT:      ex = [x1 x2]      Coordinates
%            ey = [y1 y2]
%            eq = [qx qy]      Uniform external load (global
%                               coordinate system)
%            ep = [E           Young's modulus
%                  R           Radius of curvature
%                  w_flange    Width of flange
%                  w_web       Total width of web ('walls')
%                  htot        Total height of cross section
%                  hweb        Height of web
%            flip_beam        Boolean, if set to true curvature
%                               of beam will be flipped. See chapter
%                               x for more detailed information.
%            n                Number of nodal points used
%                               for approximation (n>=2)
%
%-----
% OUTPUT:    Ke                Element stiffness matrix (global)
%            fe                Element force vector
%                               (approximation, global system)
%-----
% AUTHOR:      Erik Persson    2008-04-18
%
% Copyright (c) Erik Persson
%-----

% Element properties
E = ep(1);
R = ep(2);
w_flange = ep(3);
w_web = ep(4);
htot = ep(5);
hweb = ep(6);

% Generate coordinates
bcoord = cbeam2p(ex, ey, R, n, flip_beam);

% The curved beam element is divided into smaller parts and the
% distributed load is approximated as a number of point loads which
% attacks directly into the nodal points of the subsystem.

```



```

nbr_of_dofs = n*3;
nbr_of_elements = n-1;
K = zeros(nbr_of_dofs);
f = zeros(nbr_of_dofs,1);

% Generate local edof
ledof = zeros(n-1,7);
for i = 1:n-1
    ledof(i,:) = [i (1+(i-1)*3):1:(6+(i-1)*3)];
end;

% Generate the subsystem using cbeam2e
for p = 1:nbr_of_elements
    [Ke,fe] = cbeam2e(bcoord(p:p+1,1),bcoord(p:p+1,2), eq, ep,
flip_beam);
    [K,f] = assem(ledof(p,:),K,Ke,f,fe);
end;

% Set boundary conditions so the endings beam is fix-ended
bc=[1 0
     2 0
     3 0
     nbr_of_dofs-2 0
     nbr_of_dofs-1 0
     nbr_of_dofs 0];

% Solve the system of equations
[a,Q] = solveq(K,f,bc);

% It is now possible to "translate" the distributed load into a
equivalent
% lock force vector which means that the curved beam element can be
used as
% any beam element in the caldem toolbox.

fe = [-Q(1)
      -Q(2)
      -Q(3)
      -Q(nbr_of_dofs-2)
      -Q(nbr_of_dofs-1)
      -Q(nbr_of_dofs)];

% Calculate the stiffnes matrix for the complete element

Ke = cbeam2e(ex, ey, eq, ep, flip_beam);

%-----end-----

```

CBEAM2PDISP(EX, EY, EQ, EP, FLIP_BEAM, N, ED, MFACT)

```
function [ce_disp, mfact] = cbeam2pdisp(ex, ey, eq, ep, flip_beam, n,
ed, mfact)
```

```
% function [ce_disp, mfact] = cbeam2pdisp(ex, ey, eq, ep, flip_beam,
n, ed, mfact)
```

```
%-----
% PURPOSE
%
% This function plots the displacement of a beam section using
% (n) points. The methodology is the same as for cbeam2d1(), except
% that different end displacements are used as boundary conditions.
%
%-----
% INPUT:      ex = [x1 x2]          Coordinates
%             ey = [y1 y2]
%             eq = [qx qy]          Uniform external load (global
%                                   coordinate system)
%             ep = [E               Young's modulus
%                   R               Radius of curvature
%                   w_flange        Width of flange
%                   w_web           Total width of web ('walls')
%                   htot            Total height of cross section
%                   hweb            Height of web
%             flip_beam             Boolean, if set to true curvature
%                                   of beam will be flipped. See chapter
%                                   x for more detailed information.
%             n                     Number of nodal points used
%                                   for approximation (n>=2)
%             ed = [dx1 dy1 dr1,    Element ending displacements
%                   dx2 dy2 dr2]
%             mfact                 Magnification factor
%
%-----
% OUTPUT:     ce_disp [x1 y1       Deformed shaped
%                   .           .
%                   xn yn]
%             mfact                 Magnification factor
%
%-----
% AUTHOR:      Erik Persson    2008-04-18
%
% Copyright (c) Erik Persson
%-----
```

```
% Element properties
```

```
E = ep(1);
R = ep(2);
w_flange = ep(3);
w_web = ep(4);
htot = ep(5);
hweb = ep(6);
```

```
% Generate coordinates
bcoord = cbeam2p(ex, ey, R, n, flip_beam);
```

```
% Create local subsystem
nbr_of_dofs = n*3;
nbr_of_elements = n-1;
K = zeros(nbr_of_dofs);
```

```

f = zeros(nbr_of_dofs,1);

% Generate local edof
ledof = zeros(n-1,7);
for i = 1:n-1
    ledof(i,:) = [i (1+(i-1)*3):1:(6+(i-1)*3)];
end;

%element = 1 => p1->p2
for p = 1:nbr_of_elements
    [Ke,fe] = cbeam2e(bcoord(p:p+1,1),bcoord(p:p+1,2), eq, ep,
flip_beam);
    % Ke = beam2e(bcoord(p:p+1,1),bcoord(p:p+1,2), [210e9 0.01 8.3333e-
06]);
    [K,f] = assem(ledof(p,:),K,Ke,f,fe);
end;

% Set boundary conditions
bc=[1 ed(1)
    2 ed(2)
    3 ed(3)
    nbr_of_dofs-2 ed(4)
    nbr_of_dofs-1 ed(5)
    nbr_of_dofs ed(6)];

% Solve the system of the equations
[a,Q] = solveq(K,f,bc);

bcoord_disp = bcoord;

if (mfact == 0)
    mfact = (sqrt((ex(2)-ex(1))^2+(ey(2)-ey(1))^2)/max(abs(a)))*0.09;
end;

% Add the displacements to the original curvature
for i = 1:n
    bcoord_disp(i,1) = bcoord_disp(i,1)+a(1+((i-1)*3))*mfact;
    bcoord_disp(i,2) = bcoord_disp(i,2)+a(2+((i-1)*3))*mfact;
end;

% Plot!
plot(bcoord_disp(:,1),bcoord_disp(:,2), 'r--');

ce_disp = bcoord_disp;

%-----end-----

```

CBEAM2P(EX, EY, R, N, FLIP_BEAM)

```

function [beam_coord,xc,yc] = cbeam2p(ex, ey, R, n, flip_beam)

% function [beam_coord,xc,yc] = cbeam2p(ex, ey, R, n, flip_beam)
%-----
% PURPOSE
%
% Geometric function for generating (n) coordinates for a section
% of a circle, reaching from (x1,y1) to (x2,y2) width radius R.
% The center of such a circle is also calculated.
%
%-----
% INPUT:      ex = [x1 x2]      Coordinates
%            ey = [y1 y2]
%            R      Radius
%            n      Number of coordinates
%            flip_beam Boolean, if set to true curvature
%                   of beam will be flipped. See chapter
%                   x for more detailed information.%
%
%-----
% OUTPUT:    beam_coord = [x1 y1  Coordinates
%                          .      .
%                          xn yn]
%            xc      Center of circle (x)
%            yc      Center of circle (y)
%
%-----
% AUTHOR:      Erik Persson  2008-04-18
%
% Copyright (c) Erik Persson
%-----

% Comment: flip_beam is not yet fully implemented and should not be
used!

% Point 1, P1
x1 = ex(1);
y1 = ey(1);

% Point 2, P2
x2 = ex(2);
y2 = ey(2);

% Geometrical calculations
s = sqrt((x2-x1)^2 + (y2-y1)^2);      % |P1P2|

if(s>2*R)
    disp('Diameter error. No such circle can exist in euclidean
geometry. ');
    return;
end

x3 = (x2-x1)/2+x1;
y3 = (y2-y1)/2+y1;
m = sqrt(4*R^2-s^2)/2;                % |P3Pc|, P1P2 -> Pc

if((x2-x1) == 0)                      % Exception when y = k
    xc = x1+m
    yc = (y1+y2)/2+y1

```

```

else
    k = (y2-y1)/(x2-x1); % Slope P1P2
    v = 1/(sqrt((x2-x1)^2+(y2-y1)^2))*[x2-x1,y2-y1];
    nv = [v(2),-v(1)]; % Slope (v) P3Pc, normal
                    % vector (nv)

    if (nv(1) == 0) % Exception when k=0 to avoid
                    % division by zero
        xc = x3;
        yc = y3-m;
    else
        kc = nv(2)/nv(1); % Slope P3Pc

        % See derivation in notes from 15/1. Based on the assumption
        % that
        % |P3Pc| = m and that the equation of the line P3->Pc is known
        % since the slope is the normal of the coefficient of P1P2
        xc1 = sqrt(4*R^2-s^2)/(2*sqrt(kc^2+1)) + x3;
        xc2 = x3 - sqrt(4*R^2-s^2)/(2*sqrt(kc^2+1));

        % Because of the nature of the circle equation, two
        % centerpoints
        % are withheld.
        yc1 = kc*(xc1-x3)+y3;
        yc2 = kc*(xc2-x3)+y3;

        % Use the 'lower' of the two, unless flip_beam is true
        if ((yc1 < yc2) & (flip_beam == 0))
            xc = xc1;
            yc = yc1;
        else
            xc = xc2;
            yc = yc2;
        end
    end
end;

alpha = asin(s/(2*R))*2; % Angle P1-Pc-P2
beta = alpha/(n-1); % Angle for each piece

% Gamma denotes the angle between PcP1 and the coordinate system,
% measured from
% the slope (1,0). Depending on where on the circle P1 is located,
% gamma is
% calculated using different expressions.

if(x1<xc) % Second quarter
    gamma = pi - asin((y1-yc)/R);
else % First quarter
    gamma = asin((y1-yc)/R);
end;

beam_coord = zeros(n,2);
beam_coord(1,1) = x1; beam_coord(1,2) = y1;
beam_coord(n,1) = x2; beam_coord(n,2) = y2;

% Calculate n coordinates
for i = 1:(n-2)
    t = cos((pi-gamma)+beta*i)*R; % t is the x-part of PcPt
    xglobal = xc - t; % Conversion to global system
end;

```

```
% When solving the equation of circle, information of slope is lost
% since the square root is involved. If gamma-beta*n > pi then P is
% located in third/fourth quarter and y = -y.
if( ((gamma-beta*i) > pi) )
    yglobal = -sqrt(R^2-(xglobal-xc)^2)+yc;;
else
    yglobal = sqrt(R^2-(xglobal-xc)^2)+yc;
end;

beam_coord(i+1,1) = xglobal;
beam_coord(i+1,2) = yglobal;
end;

%-----end-----
```

CBEAM2STRESS(YS,M,V,N,R,EP)

```

function [sigma,tau,sigma_perpendicular] = cbeam2stress(ys, M, V, N,
ep)

% function [sigma,tau,sigma_perpendicular] = cbeam2stress(ys, M, V, N,
ep)
%
%-----
% PURPOSE
%
% Compute normal stress, shear stress and
% stress perpendicular to line of gravity
% for a curved beam element of box cross
% section.
%
%-----
% INPUT:      ys          Point of evaluation
%            M           Moment
%            V           Shear force
%            N           Normal force
%            ep = [E      Young's modoulous
%                  R      Radius of curvature
%                  w_flange Width of flange
%                  w_web   Total width of web ('walls')
%                  htot   Total height of cross section
%                  hweb]  Height of web
%
%-----
% OUTPUT:     sigma      Normal stress at ys
%            tau        Shear stress at ys
%            sigma_perpendicular Stress perpendicular
%                       to line of gravity at ys
%
%-----
% AUTHOR:      Erik Persson  2008-04-18
%
% Copyright (c) Erik Persson
%-----

R = ep(2);
w_flange = ep(3);
w_web = ep(4);
htot = ep(5);
hweb = ep(6);

% Constants valid for cross section
yeweb = hweb/2;
A = w_flange*(htot-hweb)+hweb*w_web;

Fflange = @(y) (w_flange.*(-y./(R+y)));
Fweb = @(y) (w_web.*(-y./(R+y)));
m = 1/A * (quadl(Fflange,-htot/2,-yeweb) + quadl(Fweb,-yeweb,yeweb) +
quadl(Fflange,yeweb,htot/2));

% Set variable 'where' depending on where to evaluate
if (ys <= htot/2 & ys > hweb/2)
    where = 'upper';
elseif (ys >= -hweb/2 & ys <= hweb/2)
    where = 'web';
elseif (ys >= -htot/2 & ys < -hweb/2)

```

```

        where = 'lower';
    else
        sprintf('Error: y out of scope');
        return;
    end;

    % Normal stress
    sigma = N/A - (M/(R*A)) * (1 + ys / (m * (R + ys)));

    % Shear stress
    switch where
        case 'web'
            tau =
                (V ./ (w_web * (R + ys).^2 * A * m)) .* (w_web .* ((yeweb.^2) ./ 8 -
                ys.^2 ./ 2) + w_flange .* (htot.^2 ./ 8 - (yeweb.^2) ./ 8));
            case 'lower'
                tau = (V ./ ((R + ys).^2 * A * m)) .* (((htot).^2 ./ 8 - ys.^2 ./ 2));
            case 'upper'
                tau = (V ./ ((R + ys).^2 * A * m)) .* (((htot).^2 ./ 8 - ys.^2 ./ 2));
    end;

    % Stress perpendicular to line of gravity
    I = w_flange * htot^3 / 12 - (w_flange - w_web) * hweb^3 / 12;
    sigma2 = @(y) (y);

    switch where
        case 'upper'
            sigma_perpendicular =
                (M / (w_flange * (R + ys) * I)) * w_flange * quadl(sigma2, ys, htot / 2);
        case 'web'
            sigma_perpendicular =
                (M / (w_web * (R + ys) * I)) * (w_web * quadl(sigma2, ys, yeweb) +
                w_flange * quadl(sigma2, yeweb, htot / 2));
            case 'lower'
                sigma_perpendicular =
                (M / (w_flange * (R + ys) * I)) * (w_flange * quadl(sigma2, yeweb, htot / 2) +
                w_web * quadl(sigma2, yeweb, -yeweb) + w_flange * quadl(sigma2, ys, -yeweb));
    end

    %-----end-----

```


CBEAM2S(EX, EY, EQ, R, N, Q, FLIP_BEAM)

```

function [es,sf] = cbeam2s(ex, ey, eq, R, n, Q, flip_beam)

% function [es,sf] = cbeam2s(ex, ey, eq, R, n, Q, flip_beam)
%-----
% PURPOSE
%
% Compute the section forces of a curved beam element
% made out of a hollow box section.
%
%-----
% INPUT:      ex = [x1 x2]      Coordinates
%            ey = [y1 y2]
%            eq = [qx qy]      Uniform external load (global
%                               coordinate system)
%            R      Radius of curvature
%            n      Number of evaluation points
%            Q      Equivalent reaction forces
%            flip_beam Boolean, if set to true curvature
%                    of beam will be flipped. See chapter
%                    x for more detailed information.
%
%-----
% OUTPUT:     es = [N V M]      Section forces
%            sf = [N V M]      Reaction forces
%
%-----
% AUTHOR:      Erik Persson    2008-04-18
%
% Copyright (c) Erik Persson
%-----

fx = Q(1);
fy = Q(2);
Mu = Q(3);
qx = eq(1);
qy = eq(2);

% Coordinates for evaluation
[bcoord,xc,yc] = cbeam2p(ex, ey, R, n, flip_beam);
es = zeros(n,3);

for i = 1:n
    xev = bcoord(i,1);
    yev = bcoord(i,2);

    x1 = xev-bcoord(1,1);
    y1 = yev-bcoord(1,2);

    % Slope
    k = -(xev-xc) / sqrt(-xev^2+2*xev*xc+R^2-xc^2);

    % Slope
    vev = ([1,k]) / (sqrt(1^2+k^2));

    % Angle between vev and (0,1)
    phi = asin(vev(2));

    % Equilibrium
    vbis = -fy - qy*x1;

```

```
nbis = -fx - qx*yl;  
V = cos(phi)*vbis - sin(phi)*nbis;  
  
N = cos(phi)*nbis + sin(phi)*vbis;  
  
M = -Mu + qy*xl^2/2 - qx*yl^2/2 - fx*yl + fy*xl;  
es(i,:) = [N V M];  
%es(i,:) = [nbis vbis M];  
end;  
  
sf = [-nbis -vbis -M];  
  
%-----end-----
```

CBEAM2WEIBULL(EX, EY, ES, EP, FLIP_BEAM)

```
function [sum_sigma, sigma_d, m, Vprisma] = cbeam2weibull(ex, ey, es,
ep, flip_beam, settings)
```

```
% function [sum_sigma, sigma_d, m, Vprisma] = cbeam2weibull(ex, ey,
es, ep, flip_beam, settings)
```

```
-----
```

```
% PURPOSE
```

```
%
```

```
% Discretization of weibull integral for stresses over a hollow
% box section. Output is used to calculate probability of collaps.
```

```
%
```

```
-----
```

```
% INPUT:      ex = [x1 x2]          Coordinates
%             ey = [y1 y2]
%             eq = [qx qy]          Uniform external load (global
%                                 coordinate system)
%             ep = [E              Young's modoulous
%                   R              Radius of curvature
%                   w_flange       Width of flange
%                   w_web          Total width of web ('walls')
%                   htot           Total height of cross section
%                   hweb]          Height of web
%             flip_beam            Boolean, if set to true curvature
%                                 of beam will be flipped. See chapter
%                                 2.1.4 for more detailed information.
%
%             settings =          Vector to define variables
%                                 used for Weibull analysis.
%                                 If set to zero default values
%                                 will be used. See chapter 2.6
%                                 for definitions of variables.
%             [m                  sigma_d
%                   Vprisma      fd_tau
%                   fd_sigma     fd_90]
```

```
%
```

```
-----
```

```
% OUTPUT:    sum_sigma            Part of weibull integral for
%                                 calculation of probability of
%                                 collaps.
%             sigma_d             Experimental value
%             m                   Experimental value
%             Vprisma             Experimental value
%
```

```
%
```

```
-----
```

```
% AUTHOR:      Erik Persson      2008-04-18
```

```
%
```

```
% Copyright (c) Erik Persson
```

```
-----
```

```
R = ep(2);
w_flange = ep(3);
w_web = ep(4);
htot = ep(5);
hweb = ep(6);
```

```
% Use this section for setup
iterations_flange = 3;
iterations_web = 10;
```

```
% Material properties
```

```

% if settings = 0, use default values, else use values defined in
'settings'

if (settings == 0)
    m = 5; % experimental value
    sigma_d = 0.74e6; % experimental value, here for 5%
    Vprisma = 0.01; % experimental value for size
of prisma

    fd_tau = 4e6; % fvd
    fd_sigma = 23e6; % ftd
    fd_90 = 0.5e6; % ft90d
else
    m = settings(1);
    sigma_d = settings(2);
    Vprisma = settings(3);
    fd_tau = settings(4);
    fd_sigma = settings(5);
    fd_90 = settings(6);
end;

k_tau = fd_tau/fd_90;
k_sigma = fd_sigma/fd_90;

h_flange = (htot-hweb)/2;
sum_sigma = 0;
nbr_of_points = length(es);
dl = (asin(sqrt( (ex(1)-ex(2))^2 + (ey(1)-ey(2))^2
)/(2*R))*R)/(nbr_of_points-1);

dv_flange = w_flange * (h_flange / iterations_flange) * dl;
dv_web = w_web * (hweb / iterations_web) * dl;
[beam_coord,xc,yc] = cbeam2p(ex, ey, ep(2), nbr_of_points, flip_beam);

line_uflange = zeros(nbr_of_points,2);
line_lflange = zeros(nbr_of_points,2);
for element_x = 1:nbr_of_points

    k = (yc-beam_coord(element_x,2))/(xc-beam_coord(element_x,1));
    alpha = pi/2-atan((yc-beam_coord(element_x,2))/(xc-
beam_coord(element_x,1)));

    if(k>0)
        flip = -1;
    else
        flip = 1;
    end;

    yp = cos(alpha) * (htot/2) * flip + beam_coord(element_x,2);
    xp = (yp-beam_coord(element_x,2))/k + beam_coord(element_x,1);
    line_uflange(element_x,:) = [xp, yp];

    yp = cos(alpha) * (-htot/2) * flip + beam_coord(element_x,2);
    xp = (yp-beam_coord(element_x,2))/k + beam_coord(element_x,1);
    line_lflange(element_x,:) = [xp, yp];

% Exception for first and last subsection (half length)
if (element_x == 1 || element_x == nbr_of_points)
    redox = 0.5;
else

```

```

        redox = 1;
    end;

    for i=1:iterations_flange
        y_flange = h_flange/(iterations_flange*2)+(i - 1) * (h_flange
        / iterations_flange);
        ys = (htot/2)-y_flange;
        [sigma,tau,sigma_perpendicular] =
cbeam2stress(ys,es(element_x,3),es(element_x,2),es(element_x,1),ep);
        sigma_comp =
sqrt(sigma_perpendicular^2+(tau/k_tau)^2+(sigma/k_sigma)^2);
        (dv_flange * redox);
        sum_sigma = sum_sigma + (sigma_comp)^m * (dv_flange * redox);

        yp = cos(alpha) * ys + beam_coord(element_x,2);
        xp = (yp-beam_coord(element_x,2))/k + beam_coord(element_x,1);
        plot(xp, yp, 'o');

    end;

    for i=1:iterations_web
        y_web = hweb/(iterations_web*2)+(i - 1) * (hweb /
iterations_web);
        ys = (hweb/2)-y_web;
        [sigma,tau,sigma_perpendicular] =
cbeam2stress(ys,es(element_x,3),es(element_x,2),es(element_x,1),ep);
        sigma_comp =
sqrt(sigma_perpendicular^2+(tau/k_tau)^2+(sigma/k_sigma)^2);
        (dv_web * redox);
        sum_sigma = sum_sigma + (sigma_comp)^m * (dv_web * redox);

        yp = cos(alpha) * ys + beam_coord(element_x,2);
        xp = (yp-beam_coord(element_x,2))/k + beam_coord(element_x,1);
        plot(xp, yp, 'o');

    end;

    for i=1:iterations_flange
        y_flange = h_flange/(iterations_flange*2)+(i - 1) * (h_flange
        / iterations_flange);
        ys = (-hweb/2)-y_flange;
        [sigma,tau,sigma_perpendicular] =
cbeam2stress(ys,es(element_x,3),es(element_x,2),es(element_x,1),ep);
        sigma_comp =
sqrt(sigma_perpendicular^2+(tau/k_tau)^2+(sigma/k_sigma)^2);
        (dv_flange * redox);
        sum_sigma = sum_sigma + (sigma_comp)^m * (dv_flange * redox);

        yp = cos(alpha) * ys + beam_coord(element_x,2);
        xp = (yp-beam_coord(element_x,2))/k + beam_coord(element_x,1);
        plot(xp, yp, 'o');

    end;
end;

plot(line_uflange(:,1),line_uflange(:,2),'g-');
plot(line_lflange(:,1),line_lflange(:,2),'g-');

end

%-----end-----

```


APPENDIX B

cbeam toolbox example

Analysis of a complex curved beam made out of hollow box section using using the cbeam toolbox extension of MATLAB/Calfem.

```

echo on
% example cbeamex1.m
%
% Purpose:
% Example of how to use the cbeam toolbox
%
close all;

% Create geometry
ex1 = [0 9];
ey1 = [0 6];
ep1 = [10400e6 10 0.4 0.2 .9 .3];

ex2 = [9 14];
ey2 = [6 3];
ep2 = [10400e6 6 0.4 0.2 .9 .3];

% Assign distributed loads
eq1 = [0 -24000];
eq2 = [0 -24000];

% Assign number of evaluation points
n = 15;

% Create stiffness matrices for the elements
[Ke1,fe1] = cbeam2dl(ex1, ey1, eq1, ep1, 0, n);
[Ke2,fe2] = cbeam2dl(ex2, ey2, eq2, ep2, 0, n);

% Element DOF
edof = [1 1 2 3 4 5 6
        2 4 5 6 7 8 9
        ];

% Assign boundary conditions
bc = [1 0
      2 0
      3 0
      8 0];

% Setup the global matrices
K = zeros(9);
f = zeros(9,1);

```

```

% Assemble the system
[K,f] = assem(edof(1,:),K,Ke1,f,fe1);
[K,f] = assem(edof(2,:),K,Ke2,f,fe2);

% Solve!
[a,Q] = solveq(K,f,bc);

% Plot the geometry
figure;
hold on;
axis square;
axis([0 20 -13 7]);
title('cbeam toolbox');

bcoord1 = cbeam2p(ex1, ey1, ep1(2), n, 0);
bcoord2 = cbeam2p(ex2, ey2, ep2(2), n, 0);

plot(bcoord1(:,1),bcoord1(:,2),'-');
plot(bcoord2(:,1),bcoord2(:,2),'-');

% Plot the deformed geometry
[ce_disp, mfact] = cbeam2pdisp(ex1, ey1, eq1, ep1, 0, n, a(1:6), 0);
cbeam2pdisp(ex2, ey2, eq2, ep2, 0, n, a(4:9), mfact);

% Compute section forces
hold on;
[es1,sf] = cbeam2s(ex1, ey1, eq1, ep1(2), n, Q(1:3), 0);
[es2,sf] = cbeam2s(ex2, ey2, eq2, ep2(2), n, sf, 0);

% Plot section forces
sfac = eldia2(ex1,[-3 -3],es1(:,3));
sfacv = eldia2(ex1,[-6 -6],es1(:,2));
sfacn = eldia2(ex1,[-9 -9],es1(:,1));

eldia2(ex2,[-3 -3],es2(:,3),[2 1],sfac);
eldia2(ex2,[-6 -6],es2(:,2),[2 1],sfacv);
eldia2(ex2,[-9 -9],es2(:,1),[2 1],sfacn);

% Compute the probability for collaps
hold on
[S1, sigma_d1, m, Vprisma] = cbeam2weibull(ex1, ey1, es1, ep1, 0, 0);
[S2, sigma_d1, m, Vprisma] = cbeam2weibull(ex2, ey2, es2, ep2, 0, 0);

% Weibull probability off collaps (must be < 1)
Q = 1/(sigma_d1 * (Vprisma/(S1+S2) )^(1/m))

```