

DYNAMICS AND DAMPING IN MOORING LINES

by

KATINKA WINGEREI

THESIS

for the degree of

MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)



*Faculty of Mathematics and Natural Sciences
University of Oslo*

June 2008

*Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo*

Foreword

This thesis defines the end of my master study in applied mathematics and mechanics at the University of Oslo. This final chapter of the two years I have been a student at the University of Oslo has been the by far most demanding. It has been a long process to derive at the final product, and the difficulties and challenges on the path have been numerous.

I would like to thank my two advisors, Associate Professor Harald Osnes at the University of Oslo and Civil Engineer Per Olav Moslet at Dr. Techn. Olav Olsen, for their input and guidance throughout this entire process.

I would also like to thank my family and good friends, and my father in particular, for their never ending love and support. I could not have done this without you.

Finally, thanks to my dear Martin, for your patience and support, day and night. Your encouragement and backup has been invaluable to me.

Katinka Wingerei
Oslo, December 2007

Summary

In a world where global warming has become an every day expression, the focus on alternative energy sources is rising. The use of wind power has long been an environment friendly option to other more polluttional energy sources. World wide there are industrial environments engaged in concept development for floating offshore windmills.

The technical challenges for risers and mooring lines continue to increase with the development towards larger water depths and ever expanding areas of use. These will normally not be built on extremely deep water, but depths between 100 – 300 meters are seen as realistic.

This thesis presents the theory needed to perform a static and dynamic analysis of a mooring line thought used on a floating windmill at 300 meters water depth. After a short description of how to build a static and dynamic element model, the analysis is carried out.

Results from both the static and dynamic element solution are presented, as well as results from an industrial simulation tool, SIMO. The dynamic solution is divided into one analysis with and one without damping.

A comparison between the different analysis and a discussion of possible deviation sum up the results from the calculations.

Table of content

1 Introduction	5
1.1 Background	5
1.2 Problem definition.....	5
1.3 Method	6
2 Theory	7
2.1 Static analysis of elastic cable line	7
2.2 Central difference method	11
2.3 Hydrodynamic damping	14
2.4 Choice of material in mooring lines	16
2.5 Maple and MATLAB	16
2.6 SIMO	17
3 Analytical calculation of static forces	19
4 Building of static element model	24
5 Building of dynamic element model	26
6 Results and comments	29
6.1 Input parameters	29
6.2 Analytical solution	30
6.3 Comparison of analytical model and static element model.....	35
6.4 Dynamic model without damping	37
6.5 Dynamic model with damping	39
6.6 SIMO model	41
7 Conclusion.....	42
References	43
Appendix A	44
Appendix B	45
Appendix C	46
Appendix D	53
Appendix E.....	54

1 Introduction

1.1 Background

In a world where global warming has become an every day expression, the focus on alternative energy sources is rising. Here in Norway we are very privileged considering the wealth in oil resources we possess. But nothing lasts forever, neither the black gold. A steady growth in population and an increase in standard of living lead to a bigger need for energy. The use of wind power has long been an environment friendly option to other more pollutional energy sources.

When the windmills are placed off shore they are removed away from buildings, avoiding health damages due to for example noise. The placement requires less demands concerning damping of larm and less demands concerning design and aesthetic. This makes development of the mills less expensive. In addition the energy potential increases on deep water because the average wind speed is higher away from the coast.

At the same time one enables area on land to other activities, whether it is agriculture, forestry or as recreational areas. Windmills are an environmental alternative to fossil based production of energy, as the production of electricity by the help of windmills releases less CO₂. Wind is a renewable energy resource, and wind power helps save the environment without increasing the cost of the power.

By developing a concept where formerly unexploited sea areas can be used to produce electricity, one creates a potential to ensure more nations world wide economical access to an endless energy source without pollution and with minimal environmental damage. It can help increase living standards because the need to import energy will decrease. At the same time there is a growing need for and demand for renewable energy.

1.2 Problem definition

The technical challenges for risers and mooring lines continue to increase with the development towards larger water depths and ever expanding areas of use. Floating windmills are only one example of a new area of use. These will normally not be built on extremely

deep water, but depths between 100 – 300 meters are seen as realistic. This thesis will deal with the static and dynamic behaviour of mooring lines at these kinds of depths. The thesis is completed in cooperation with Dr. Techn. Olav Olsen as.

Problems to be dealt with are

- A) a description of necessary theory
- B) analytical calculations
- C) numerical calculations with and without damping
- D) validation of results compared to results from industrial simulation tool

1.3 Method

Chapter 1 includes a brief review over the purpose of this thesis.

Chapter 2 presents the theory necessary to perform the different kind of analysis. Here the theory behind a static solution of the line is given as well as methodology and damping needed to perform a numerical analysis. In the section for damping theory the hydrodynamic damping is the one represented. Initially also structural damping was to be implemented, but with agreement from both advisors, this is left out due to time restrictions.

Chapter 3 holds an analytical calculation of the static case, estimating forces and shape of the line.

Chapter 4 and 5 describes how the element model of the static and dynamic solution is built.

Chapter 6 is a presentation of the results derived with the methods described in chapter 3, 4 and 5. This includes comparison between methods and comments to the different results.

Chapter 7 consist of a brief account of the results, and a final comment.

2 Theory

2.1 Static analysis of elastic cable line

To prevent a floating offshore windmill from drifting away it needs to have lines that tie it to the seabed. At the end of each line there is an anchor. There are various types of anchors, but most used are conventional plough anchors that can respond only to horizontal forces.

Therefore it is common that a significant part of the anchor line lie on the seabed. The anchor line is made up of chain, wire rope or a combination of both. As the unit moves in response to unsteady environmental loads, the tension in the cables changes due to varying cable geometry. In the following it will be shown how to perform a static analysis of an anchor line (Faltinsen 1990).

A picture of an anchor line is shown in Figure 2.1. The cable lies in the x - z -plane. We assume a horizontal sea bed. We neglect bending stiffness which is a good approximation for chains. It is also appropriate for wires with a large radius of curvature. We will neglect structural dynamic effects in the line.

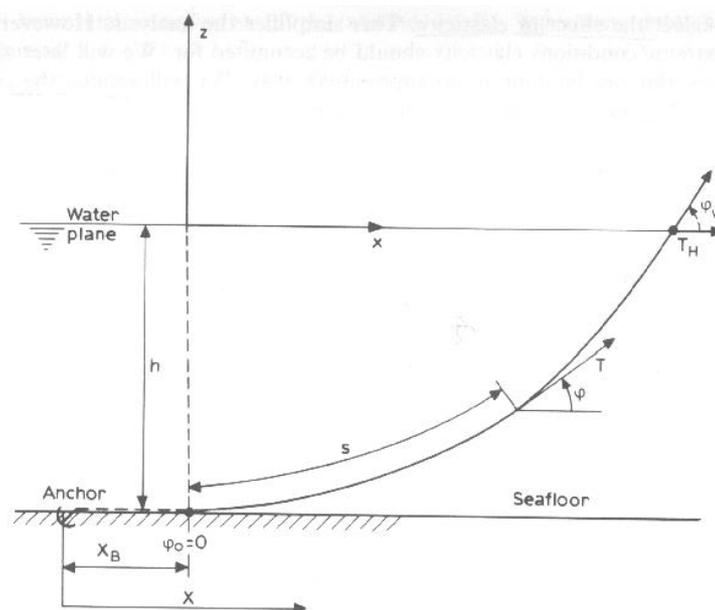


Figure 2.1: Anchor line.

Figure 2 shows one element of the cable line. Forces D and F acting on the element are the mean hydrodynamic forces per unit length in the normal and tangential direction respectively. w is the weight per unit length of the line in water, A is the cross-sectional area of the cable

line, E is the elastic modulus and T is the line tension. Because w is the weight in water, this introduces correction forces, $-\rho gAz$ and $-\rho gAz - \rho gAdz$ at the ends of the element.

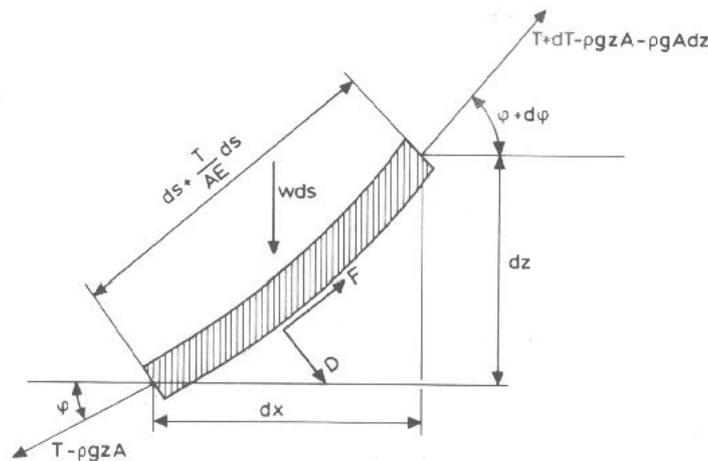


Figure 2.2: Element of anchor line.

It is not necessarily immediately clear why respectively $\rho g z A$ and $\rho g z A + \rho g A dz$ is applied as forces on the ends. This is not common in other literature. What is implicit alleged is that the sum of the hydrostatic forces on the line have a vertical resultant equal to the weight of the submerged volume of fluid (confer with the definition of w). This is not correct, but would be correct if one also had water pressure on the end surface. This is the reason for the correction applied to the end surface.

In this way it is possible to calculate correctly the hydrostatic forces on the element. Static equilibrium at the element in Figure 2.2 gives us that

$$dT - \rho g A dz = [w \sin \phi - F(1 + \frac{T}{AE})] ds \quad (2.1)$$

$$T d\phi - \rho g A z d\phi = [w \cos \phi + D(1 + \frac{T}{AE})] ds \quad (2.2)$$

These equations are non-linear and it is in general not possible to find an explicit solution. However, for many operations it is a good approximation to neglect the effect of the current forces F and D . We will assume the cable line has constant weight per unit length.

By introducing

$$T' = T - \rho g z A \quad (2.3)$$

we can write

$$dT' = w \sin \phi ds \quad (2.4)$$

$$T' d\phi = w \cos \phi ds \quad (2.5)$$

By dividing these two equations we see that

$$\frac{dT'}{T'} = \frac{\sin \phi}{\cos \phi} d\phi \quad (2.6)$$

i.e.

$$T' = T_0 \frac{\cos \phi_0}{\cos \phi} \quad (2.7)$$

By integrating eq. (2.5) we find that

$$s - s_0 = \frac{1}{w} \int_{\phi_0}^{\phi} \frac{T_0'}{\cos \theta} \frac{\cos \phi_0}{\cos \theta} d\theta = \frac{T_0' \cos \phi_0}{w} [\tan \phi - \tan \phi_0] \quad (2.8)$$

In a static case the horizontal forces in the line will always be the same. This can easily be shown by static equilibrium of the entire system. The tension is changing due to the weight, length and geometry of the cable. This means that only the vertical forces change along the line. Due to this it is only interesting to calculate the forces on the part of the line that goes from the seabed to the water plane. It is therefore naturally to choose s_0 and ϕ_0 to be zero.

This simplifies eq. (2.7) and eq. (2.8) down to respectively

$$T' = \frac{T_0'}{\cos \phi} \quad (2.9)$$

$$s = \frac{T_0'}{w} \tan \phi \quad (2.10)$$

To find the x – and z – coordinates we have to use the following relationship between the stretched length dp and unstretched length ds of a cable element

$$dp = ds \left(1 + \frac{T}{AE} \right) \quad (2.11)$$

This means

$$\frac{dx}{ds} = \cos \phi \left(1 + \frac{T}{AE} \right) \approx \cos \phi \left(1 + \frac{T'}{AE} \right) = \cos \phi + \frac{T_0'}{AE} \quad (2.12)$$

$$\frac{dz}{ds} = \sin \phi \left(1 + \frac{T}{AE} \right) \approx \sin \phi \left(1 + \frac{T'}{AE} \right) = \sin \phi + \frac{w}{AE} s \quad (2.13)$$

The last term on the right hand side of eq. (2.13) follows by combining eq. (2.9) with eq. (2.10). We will now integrate eq. (2.12) and (2.13) from the seabed up to the attachment point of the cable line to the windmill. For simplicity we will assume that the attachment point is at $z = 0$. By integrating eq. (2.13) we find that

$$h = \frac{T_H}{w} \left(\frac{1}{\cos \phi_w} - 1 \right) + \frac{1}{2} \frac{w}{AE} l_s^2 \quad (2.14)$$

We know that $\cos \phi_w = \frac{T_H}{\sqrt{T_H^2 + T_Z^2}}$. By implementing this into eq. (2.14) we get an expression

for the horizontal tension at the endpoint of the anchor line

$$T_H = \frac{T_Z^2 - \left(wh - \frac{1}{2} \frac{w^2 l_s^2}{AE} \right)^2}{2 \left(wh - \frac{1}{2} \frac{w^2 l_s^2}{AE} \right)} \quad (2.15)$$

We know from a static equilibrium that

$$T_Z = w l_s \quad (2.16)$$

We should perhaps also notice that the total tension at the attachment point is simply given by

$$T = \sqrt{T_H^2 + T_Z^2} \quad (2.17)$$

By integrating eq. (2.12) in the same way we find that

$$x = \frac{T_H}{w} \log \left(\frac{\sqrt{T_H^2 + T_Z^2} + T_Z}{T_H} \right) + \frac{T_H}{AE} l_s \quad (2.18)$$

By adding the part of the line that lies on the seabed to this parameter, we get the distance needed between the anchor and the windmill (Figure 2.3).

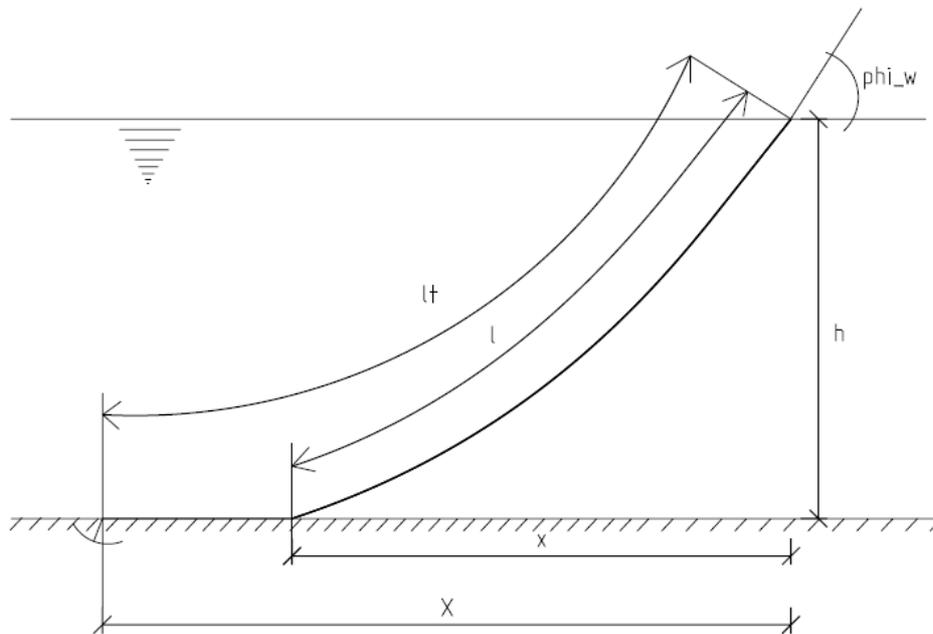


Figure 2.3: Distance between anchor and windmill.

2.2 Central difference method

An anchor line will experience different kinds of external forces. Waves, wind and current are responsible for dynamic excitation of the cable line. To determine the forces resulting from the dynamic excitation we need to perform a dynamic analysis. The following theory is taken from Chopra (2007) and Osnes (2007).

The forces acting on the mass at some instant of time are shown in Figure 2.4. These include the external force $p(t)$, the elastic or inelastic force f_s , and the damping resisting force f_D . The external force, the displacement $u(t)$, the velocity $\dot{u}(t)$ and the acceleration $\ddot{u}(t)$ is taken to be positive in the direction of the x -axis. The elastic and damping forces are shown acting in the opposite direction because they are internal forces that resist the deformation and velocity, respectively.

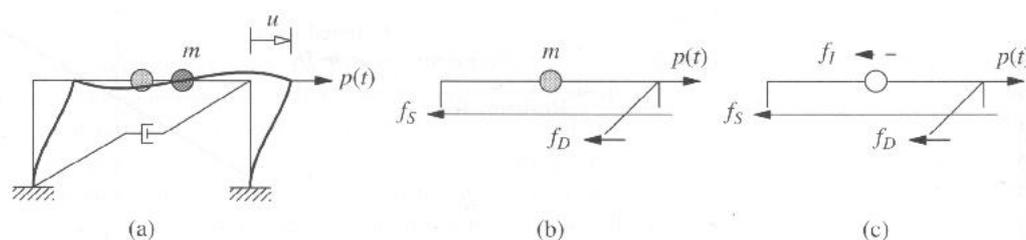


Figure 2.4: Forces on idealized one-story frame

The resultant force along the x -axis is $p - f_S - f_D$ and Newton's second law of motion gives

$$p - f_S - f_D = m\ddot{u} \quad (2.19)$$

or

$$m\ddot{u} + f_D + f_S = p(t) \quad (2.20)$$

For a linear system the relationship between the lateral force f_S and resulting deformation u is linear, that is,

$$f_S = ku \quad (2.21)$$

where k is the lateral stiffness of the system. The damping force f_D is related to the velocity \dot{u} across a linear viscous damper by

$$f_D = c\dot{u} \quad (2.22)$$

After substituting eqs. (2.21) and (2.22), eq. (2.20) becomes

$$m\ddot{u} + c\dot{u} + ku = p(t) \quad (2.23)$$

This is the equation of motion governing the deformation or displacement $u(t)$ of the idealized structure of Figure 2.4, assumed to be linearly elastic, subjected to an external dynamic force $p(t)$.

Analytical solution of the equation of motion for a single-degree-of-freedom system is usually not possible if the excitation – applied force $p(t)$ varies arbitrarily with time or if the system is nonlinear. Such problems can be tackled by numerical time-stepping methods for integration of differential equations. Stepping from time i to $i + 1$ is usually not an exact procedure. Many approximate procedures are possible that are implemented numerically. The three important requirements for a numerical procedure are (1) convergence - as the time step decreases, the numerical solution should approach the exact solution, (2) stability – the numerical solution should be stable in the presence of numerical round-off errors, and (3) accuracy – the numerical procedure should provide results that are close enough to the exact solution. The central difference method is based on a finite difference approximation to the time derivatives of displacement (i.e. velocity and acceleration). It is explicit for systems with one degree of freedom. It is generally implicit for multi-degree-of-freedom systems, but explicit if the mass- and damping matrices are diagonal.

Taking constant time steps, $\Delta t_i = \Delta t$, the central difference expressions for velocity and acceleration at time i are

$$\dot{u}_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta t} \quad (2.24)$$

$$\ddot{u}_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta t)^2} \quad (2.25)$$

Substituting these approximate expressions for velocity and acceleration into eq. (2.23) gives

$$m \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta t)^2} + c \frac{u_{i+1} - u_{i-1}}{2\Delta t} + ku_i = p_i \quad (2.26)$$

In this equation u_i and u_{i-1} are known from implementation of the procedure for the preceding time steps. Transferring these known quantities to the right side leads to

$$\left(\frac{m}{(\Delta t)^2} + \frac{c}{2\Delta t} \right) u_{i+1} = \left(\frac{2m}{(\Delta t)^2} - k \right) u_i - \left(\frac{m}{(\Delta t)^2} - \frac{c}{2\Delta t} \right) u_{i-1} + p_i \quad (2.27)$$

In eq. (2.27) the known displacements u_i and u_{i-1} are used to compute u_{i+1} . Thus u_0 and u_{-1} are required to determine u_1 . This means that in order to set up the algorithm, the case $i = 0$ needs to be treated separately. For $i = 0$ the eq. (2.27) can be written as

$$\hat{K}u_1 = \hat{K}_1u_0 - \hat{K}_2u_{-1} + p_0 \quad (2.28)$$

The specified initial displacement u_0 is known by

$$u(0) = u_{stat} \quad (2.29)$$

u_{stat} being the initial displacement from a static solution, and

$$\dot{u}(0) = 0 \quad (2.30)$$

meaning that the system is at rest before the dynamic load case is being applied. To determine u_{-1} , we specialize eq. (2.24) for $i = 0$ to obtain

$$\dot{u}(0) = \frac{u_1 - u_{-1}}{2\Delta t} \quad (2.31)$$

By combining eqs. (2.30) and (2.31) we get that

$$u_{-1} = u_1 \quad (2.32)$$

We now implement eq. (2.32) into eq. (2.28) which gives us that

$$\hat{K}u_1 = \hat{K}_1u_{stat} - \hat{K}_2u_1 + p_0 \quad (2.33)$$

This is equal to

$$\left(\frac{m}{(\Delta t)^2} + \frac{c}{2\Delta t}\right) + \left(\frac{m}{(\Delta t)^2} - \frac{c}{2\Delta t}\right)u_1 = \left(\frac{2m}{(\Delta t)^2} - k\right)u_{stat} + p_0 \quad (2.34)$$

or

$$\left(\frac{2m}{(\Delta t)^2}\right)u_1 = \left(\frac{2m}{(\Delta t)^2} - k\right)u_{stat} + p_0 \quad (2.35)$$

which defines the beginning of the algorithm.

Systems without damping

For a system without damping, the parts of the equation that include damping is neglected.

This gives, on a general form

$$\left(\frac{m}{(\Delta t)^2}\right)u_{i+1} = \left(\frac{2m}{(\Delta t)^2} - k\right)u_i - \left(\frac{m}{(\Delta t)^2}\right)u_{i-1} + p_i \quad (2.36)$$

The beginning of the algorithm can be calculated in the same way as for a system with damping.

Stability

A central differential method is conditionally stable. The specific requirement for stability is

$$\frac{\Delta t}{\Delta x} < \frac{1}{\pi} \quad (2.37)$$

This means that it is necessary to chose $\frac{\Delta t}{\Delta x} < \text{critical_value}$ for the numerical solution to act

stable. If the discretization in space is coarse (meaning we have few elements) the behaviour of the line is normal. When increasing the number of elements (i.e. the element length is decreased), usually Δt needs to be reduced accordingly to ensure stable behaviour.

2.3 Hydrodynamic damping

An object moving through a gas or liquid experiences a force in direction opposite to its motion. In fluid dynamics, drag (sometimes called resistance) is the force that resists the movement of a solid object through a fluid (a liquid or gas). Drag is made up of friction forces, which act parallel to the object's surface plus pressure forces, which act in a direction

perpendicular to the object's surface. For a solid object moving through a fluid, the drag is the sum of all the aerodynamic or hydrodynamic forces in the direction of the movement:

$$F_D = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot v^2 \quad (2.38)$$

where C_D is the drag coefficient, ρ is the density of the medium, v is the flow speed, and A is the cross-sectional area.

In order to dampen the motion of the line one is required to calculate these forces and apply them to the line in the opposite direction of what they are acting (Figure 2.5).

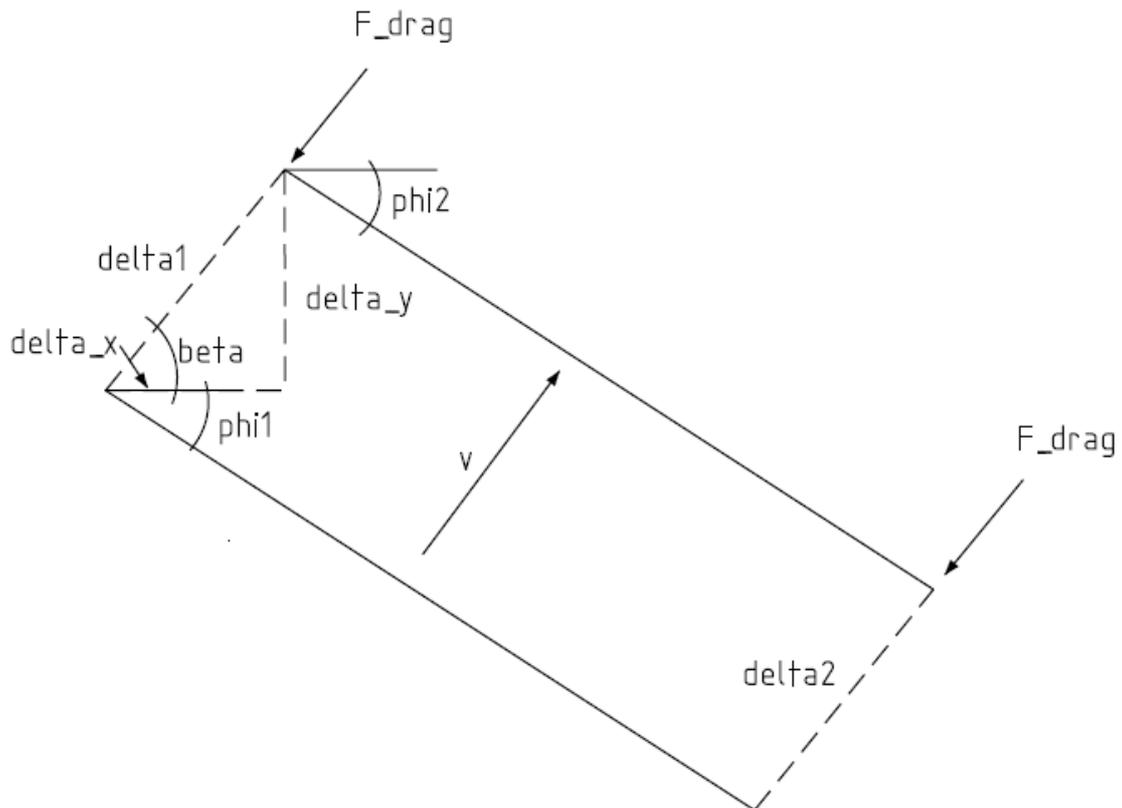


Figure 2.5: Drag force.

2.4 Choice of material in mooring lines

A mooring line or anchor line is usually made up of more than one material. One always tries to optimize the balance between economics and material properties. This demands knowledge about the different materials. The amount of research done on wire rope far outweighs that on synthetic fibre ropes. Research has shown that synthetic fibre ropes have better endurance and higher strength than steel wire ropes. Synthetic cables are much lighter and can absorb dynamic motions through slight elongation without large amounts of dynamic tension better than steel cables. This makes fibre cables more suitable in certain situations. Other advantages of synthetic fibre ropes are ease of handling and safety. Synthetic fibre is a popular material due to their excellent vibration damping, flex fatigue and internal fibre-friction characteristics (Wakecord 2007). Minimal moisture absorption and excellent flex/fold/abrasion/impact resistance are other important material advantages (Swicofil 2008).

The three most common conventional rope materials are nylon, polyester and propylene. Nylon has the lowest E-modulus but is the strongest of all when dry. Polyester is very durable under cyclic tensile fatigue loading and has a high E-modulus. Polypropylene is the weakest of the three, but is most commonly found in marine applications. It is inexpensive and floats (Pearson 2002).

Steel wire ropes behave similarly to fibre ropes under tension loading through the elastic region. Steel cables have the ability to resist large axial tensile loads as compared to torsional or bending loads. One of the maybe most attractive properties of ropes is the very high strength-to-weight ratio (Pearson 2002). Steel also has a higher elasticity modulus, which makes it stiffer than most fibre cables. One normally wishes to keep the windmill or body as stationary as possible. This requires demands to the stiffness of the mooring lines. The higher E-modulus and stiffness of steel can therefore be used in a beneficial way.

2.5 Maple and MATLAB

Maple is a general-purpose commercial mathematics software package. Since 1988, it has been developed and sold commercially by Waterloo Maple Inc. (also known as Maplesoft). Users can enter mathematics in traditional mathematical notation. Custom user interfaces can

also be easily created. There is support for both numeric and symbolic computation, as well as visualization.

MATLAB (short for "MATrix LABoratory") is a numerical computing environment and programming language. Created by The MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages.

2.6 SIMO

SIMO is short for Simulation of Marine Operations. It is a time domain simulation program for multibody systems allowing non-linear effects to be included in the wave-frequency range. SIMO is owned, developed and maintained by MARINTEK, - marketing and sale is handled by DNV Software. Flexible modelling of station keeping forces and connecting force mechanisms (anchor lines, ropes, and thrusters) is included.

The results from the program are presented as time traces, statistics and spectral analysis of all forces and motions of all bodies in the analysed system. SIMO is a modular and interactive computer program with batch processing options. Essential features are:

- Flexible modelling of multibody systems
- Non-linear time domain simulation of wave-frequency as well as low-frequency forces.
- Environmental forces due to wind, waves and current.
- Passive and active control forces.
- Interactive or batch simulation.

In the following a short description of the most important system functions and features and a layout of the SIMO program system and file communication between modules (Figure 2.6) will be shown.

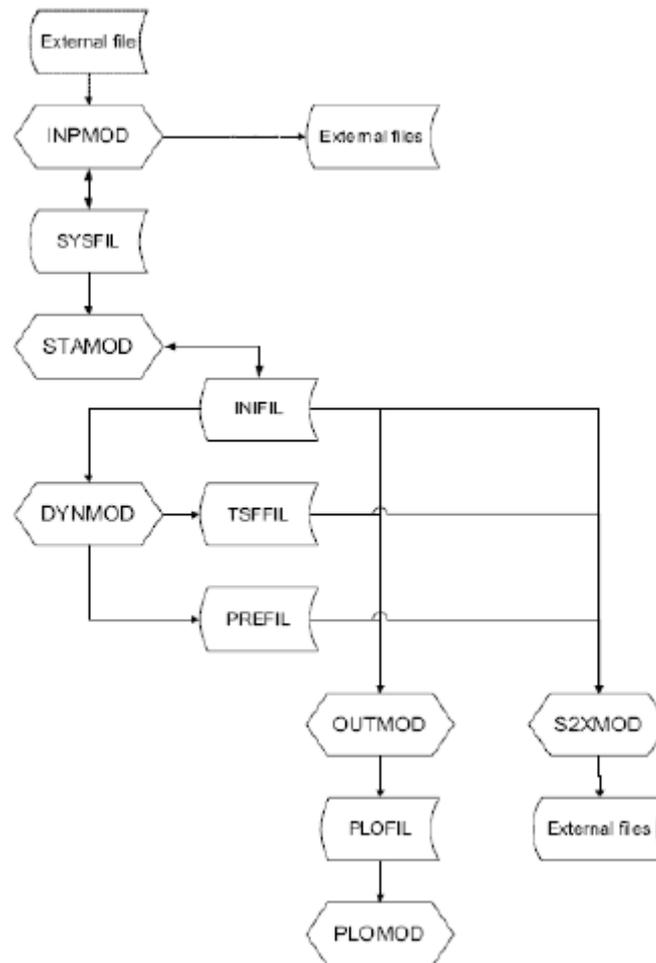


Figure 2.6: Flow diagram of SIMO (SIMO User Manual).

INPMOD, input data manipulation

This input module accepts external data, e.g. a model from a hydrodynamic program such as WADAM, and modifies a file for further use in SIMO.

STAMOD, initial condition and static equilibrium

Here the file from INPMOD is read, and new parameters are added. It is possible to apply the external forces that work on the model and from this calculate static equilibrium, natural periods and oscillation modes for the system. STAMOD defines the initial positions for dynamic simulation and writes them to a new file to be used in DYNMOD.

DYNMOD, dynamic response calculation

This module calculates the dynamic responses of the model. Here you apply parameters like wind and waves. This is where the time integration takes place. Main analysis parameters, method parameters, storage parameters and integration parameters must be specified.

OUTMOD, output module

This is where the time series file from DYNMOD is being read. This module generates plots and presents results from the time domain simulation.

3 Analytical calculation of static forces

Anchor lines can be made up of different types of material. A combination of chain and rope is common. The ropes are available in constructions from steel, natural fibre and synthetic fibres. In this case we have an anchor line that is made up of both chain and fibre-rope. The benefits of segmenting an anchor line is that one gets a heavy cable at the bottom (i.e. chain), and a light line close to the water surface. This gives greater stiffness and lighter anchor lines, than the use of chain or wire alone (Faltinsen 1990). As a line made up entirely of chain would weight very much, the use of fibre in parts of the line is beneficial. But the fibres design life is critical shortened if it is in contact with the sand on the seabed, as this will destroy the fibres.

The challenge is to optimize the point where the chain ends and the fibre starts. Figure 3.1 shows the entire line, where the line is segmented into two materials, 1 being chain and 2 being fibre. The following parameters are known or assumed: weight of chain and fibre (w_1 and w_2), the total height h , the total length from the anchor to the windmill X , the total length of the line l , the total length of chain l_c , the total length of fibre l_f and the tension at the end of the line T . Because we know the tension at the water plane it is possible to calculate the horizontal force G_x from a static equilibrium. By use of Pythagoras it is not difficult to find G_y and the angle φ_w . We also know the E-modulus and the cross section area of both the chain and the fibre.

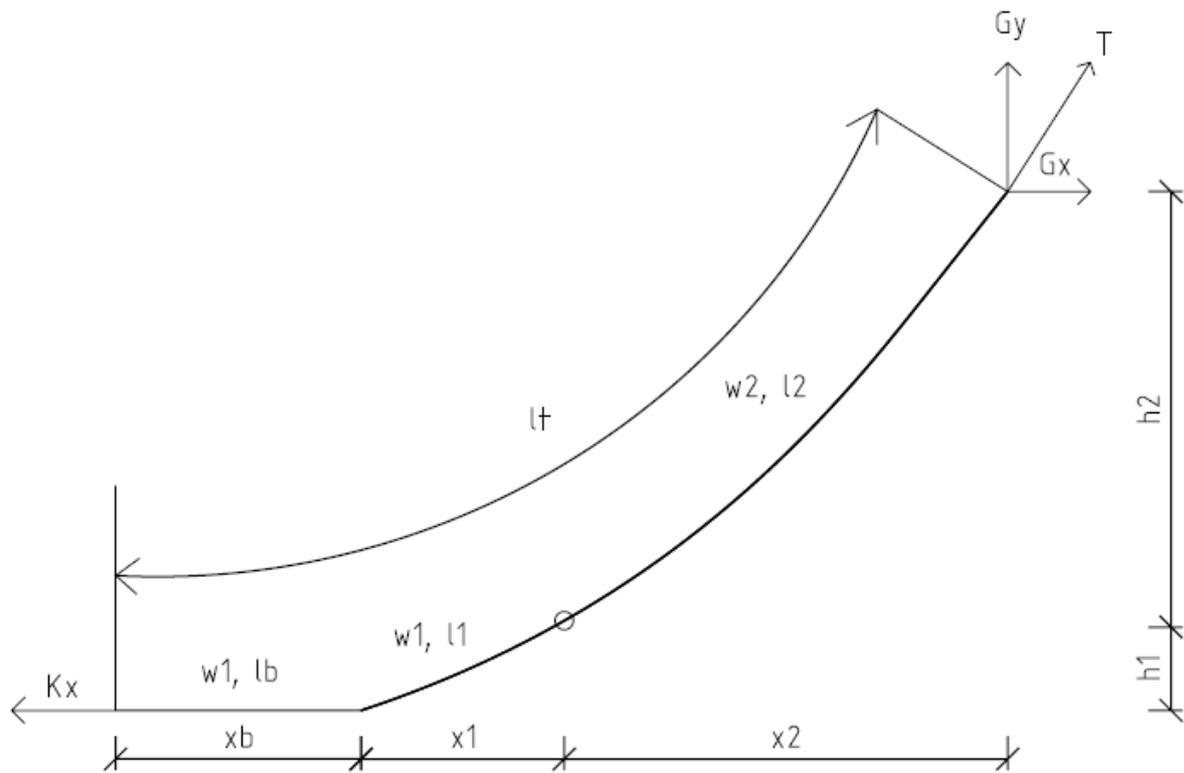


Figure 3.1: Entire anchor line.

In Figure 3.2 the line is shown divided into two segments. Segment number one is in this case wire rope, and segment number two is fibre line.

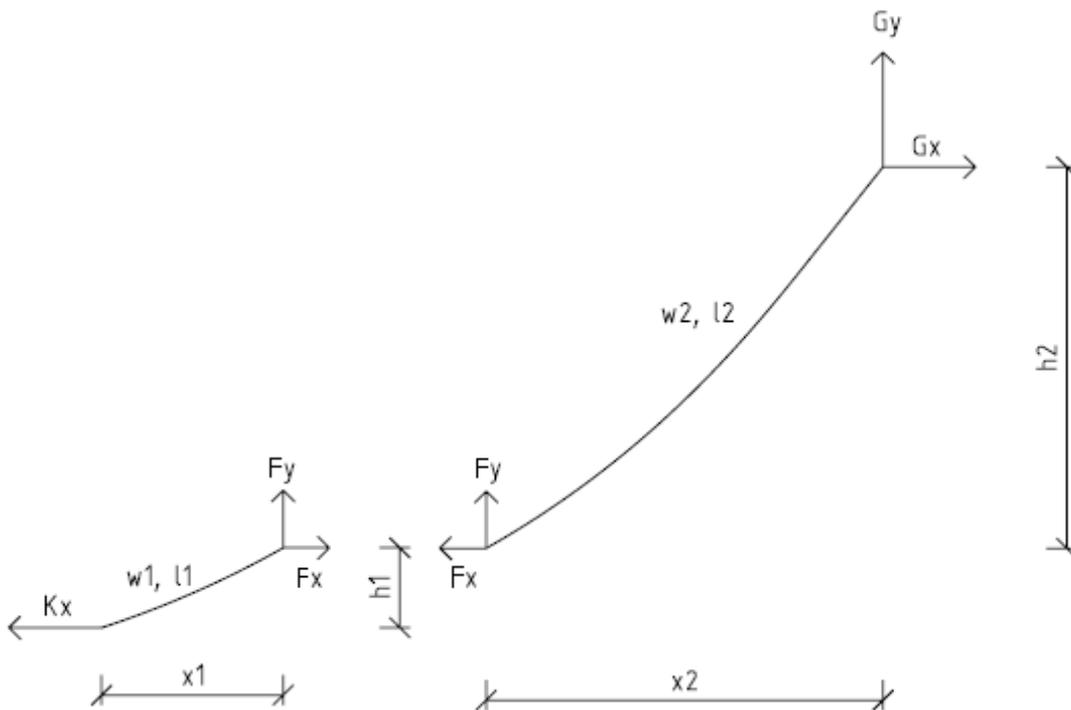


Figure 3.2: Anchor line divided into two segments.

For the chain part of the line we can look at the line as a single line made up of one material and take advantage of the static analysis shown in chapter 2.1. Static equilibrium of the lower part gives us that

$$K_x = F_x \quad (3.1)$$

$$F_y = w_1 l_1 \quad (3.2)$$

From eq. (2.15) we get that

$$F_x = \frac{F_y^2 - \left(w_1 h_1 - \frac{1}{2} \frac{w_1^2 l_1^2}{A_1 E_1} \right)^2}{2 \left(w_1 h_1 - \frac{1}{2} \frac{w_1^2 l_1^2}{A_1 E_1} \right)} \quad (3.3)$$

We now have the horizontal force in the line. As described briefly in chapter 2.1 horizontal forces are the same through the entire line.

The expression used to find F_x is based on the assumption that the line starts from the seabed, i.e. both the angle and the height at the lower end is zero. For the upper part of our line, this is not the case. It is not possible to use eq. (2.15) as it stands. We have to derive a new expression were we take into consideration that neither the angle nor the height at the lower end is zero. This means that eqs. (2.7) and (2.8) are still valid:

$$T' = T_0 \frac{\cos \phi_0}{\cos \phi} \quad (3.4)$$

$$s - s_0 = \frac{T_0 \cos \phi_0}{w} (\tan \phi - \tan \phi_0) \quad (3.5)$$

By combining these two equations, eq. (2.12) and eq. (2.13) becomes the following

$$\frac{dx}{ds} = \cos \phi \left(1 + \frac{T}{AE} \right) \approx \cos \phi \left(1 + \frac{T'}{AE} \right) = \cos \phi \left(1 + \frac{T_0 \cos \phi_0}{AE \cos \phi} \right) = \cos \phi + \frac{T_0}{AE} \cos \phi_0 \quad (3.6)$$

$$\begin{aligned} \frac{dz}{ds} &= \sin \phi \left(1 + \frac{T}{AE} \right) \approx \sin \phi \left(1 + \frac{T'}{AE} \right) = \sin \phi \left(1 + \frac{T_0 \cos \phi_0}{AE \cos \phi} \right) = \sin \phi + \frac{T_0}{AE} \tan \phi \cos \phi_0 \\ &= \sin \phi + \frac{sw}{AE} \frac{\tan \phi}{\tan \phi - \tan \phi_0} \end{aligned} \quad (3.7)$$

We now integrate eq. (3.7) in the same way as we integrated eq.(2.13), and we thereby get

$$z - z_0 = \frac{T_0' \cos \phi_0}{w} \left[\frac{1}{\cos \phi_w} - \frac{1}{\cos \phi_0} \right] + \frac{1}{2} \frac{w}{AE} [s^2 - s_0^2] \quad (3.8)$$

As $\cos \phi_w$ is the angle the line deviates from the water plane, we know that

$$\cos \phi_w = \frac{T_H}{\sqrt{T_H^2 + T_Z^2}} \quad (3.9)$$

We also know the horizontal force in the line, which enables us to say that

$$T_0 \cos \phi_0 = Fx \quad (3.10)$$

where T_0 is the tension at the start of the upper part of the anchor line. By combining eq. (3.9) and eq. (3.10) with eq. (3.8) we get a new expression for the horizontal force at the end of the line. This expression is valid only for the fibre part of the line, as these parts has different initial conditions at the chain part.

$$T_H = \frac{T_Z}{\sqrt{\left(\frac{w_2 h_2}{F_x} - \frac{1}{2} \frac{w_2^2 l_2^2}{A_2 E_2 Fx} + \frac{1}{\cos \phi_0} \right)^2 - 1}} \quad (3.11)$$

By doing a static equilibrium of the upper part of the line we get that

$$G_x = F_x \quad (3.12)$$

$$G_x = T_{H_upper} \quad (3.13)$$

$$G_y = F_y + w_2 l_2 = w_1 l_1 + w_2 l_2 \quad (3.14)$$

Eq. (3.12) and eq. (3.14) contain the unknown value l_1 , and eq. (3.13) contains both unknown's l_1 and h_1 . If we manage to determine these values, it is possible to find the coordinates for the point where the line makes a shift in material. This can be done by combining the equations for the forces in horizontal direction. We see that

$$F_x = G_x = T_{H_upper} \quad (3.15)$$

Because both F_x and G_x only contain one unknown, l_1 , the logical thing to do would be to solve this equation with respect to l_1 , and then solve for h_1 . Implementation of this equation in both MATLAB and Maple only gives the answer “solutions may be lost” or “solution not found”. We therefore need a set of two equations to find our two unknowns.

We can use eq. (3.15) to get a set with two equations. The important thing to consider is that we need to have both l_l and h_l in both equations. This means that T_{H_upper} needs to be represented in both equations, as it is the only expression that holds h_l

$$F_x = T_{H_upper}, G_x = T_{H_upper} \quad (3.16)$$

Eq. (3.16) can be solved by the use of Maple. We then get values for both unknowns.

To find the shape of the line we use an equation for an unsymmetrical catenary cable taken from Fridtjov Irgens (1992)

$$y(x) = \frac{q}{2F_x}(x^2 - Lx) + \frac{h}{L}x \quad (3.17)$$

where L is the distance along the seabed.

By applying different material parameters to different parts of the line we get the correct shape of the line for different cases. This is done by assuming one material up to a certain point, and another material from that point and up to the waterbed. It, however, requires that we know the coordinates for the shift in material.

From earlier calculations we know the length and height (y -coordinate) of this point, but we also need to find the length along the seabed (x -coordinate). This can be found using the catenary formula (Calvert 2000)

$$y = \frac{H}{w} \left(\cosh \frac{wx}{H} - 1 \right) \quad (3.18)$$

Since x is our unknown, we solve eq. (3.18) with respect to x

$$x = \frac{H}{w} \cosh^{-1} \left(\frac{yw}{H} + 1 \right) \quad (3.19)$$

Translated into our specific problem this reads

$$x_1 = \frac{G_x}{w_1} \cosh^{-1} \left(\frac{h_1 w_1}{G_x} + 1 \right) \quad (3.20)$$

We now have all we need to plot the shape of the line.

4 Building of static element model

It is necessary to build a static model of the line before performing a dynamic analysis. The static model will work as one of the initial conditions for the dynamic part. The static model is built upon the data received from the analytical solution in chapter 3. This means that the geometrical boundaries are known.

The first step in building an element model is to define how many elements the line should consist of. From the analytical solution we find the total horizontal force in the line. We call this force F_x (Figure 4.2). To start of the routine we need to find the vertical force at the beginning of the line as well. For the part of the line that lies on the seabed, the line only has an axial force. This means that it is appropriate to start the calculations at the node were the line lifts of the ground. This is also convenient as we now the coordinates for this point. For this node the vertical force F_y is expressed by

$$F_{y1} = F_{y0} + \frac{1}{2} w_0 \cdot l_{e1} \quad (4.1)$$

where l_{e1} is the length of the element. For simplicity this node will in the further be called node 1. Here the sub index 0 refers to the element before node one, i.e. the last element that lie on the seabed. The last term on the right hand side of the equation is due to lumping of the element mass. The element has an evenly distributed mass throughout an actual structure, but it can be idealized as lumped at the nodes of the discretized structure; usually, such a lumped-mass idealization is satisfactory. Lumping of the mass could not have been done had the mass not been evenly distributed.

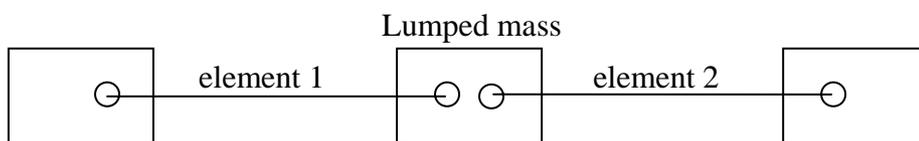


Figure 4.1: Lumping of mass at structural nodes.

The lumped mass at a node is determined from the portion of the weight that can reasonably be assigned to the node. Each structural element is replaced by point masses at its two nodes, with the distribution of the two masses being determined by statics. The lumped mass at a

node of the structure is the sum of the mass contributions of all the structural elements connected to the node (Chopra 2007).

In this case it means that half of the mass from each element will be distributed to the node between the elements (Figure 4.1). Because we start our calculations at node 1, this node will only have half the mass from the element that lifts of the ground.

It is now possible to find the line tension T by using Pythagoras

$$T = \sqrt{F_x^2 + F_y^2} \quad (4.2)$$

It is easy to use the forces that act on node 1 to find the angle φ_1 . We now have enough information to estimate the coordinates for node 2. We use the relationship between the stretched and unstretched length from eq.(2.11), and get that the coordinates are respectively

$$x_2 = x_1 + element_length_1 \left(1 + \frac{T_1}{A_1 E_1} \right) \cos \varphi_1 \quad (4.3)$$

$$y_2 = y_1 + element_length_1 \left(1 + \frac{T_1}{A_1 E_1} \right) \sin \varphi_1 \quad (4.4)$$

with T_1 , A_1 and E_1 being the tension, cross section area and elasticity module for each element. From here the routine starts from the top again. The horizontal force is still the same; the only thing changing is the vertical force, as this is a result of the weight of the material.

Static equilibrium gives us that

$$F_{y2} = F_{y1} + \left(\frac{1}{2} w_1 + \frac{1}{2} w_2 \right) element_length_1 \quad (4.5)$$

As described above, lumping of the mass now gives us a mass contribution from the elements on both sides of the node. This gives us enough information to start a general routine that goes through the entire set of elements and calculates the shape of the line with the given material parameters.

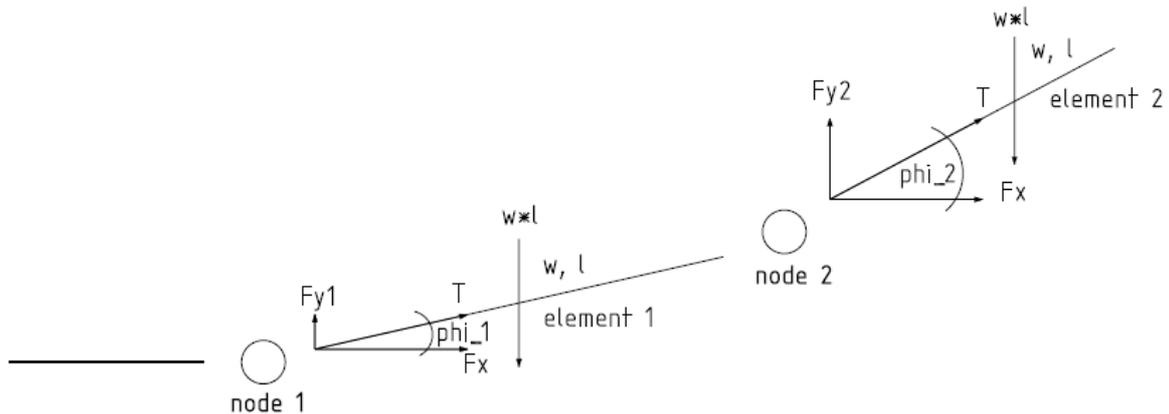


Figure 4.2: Part of anchor line divided into elements.

5 Building of dynamic element model

An anchor line will be exposed to waves, current and wind in various strength. To study the impact on the line from an external force, it is necessary to build a dynamic element model. This, as the static element model, can be done in MATLAB.

The first step would be to divide our timeline into time steps. This is needed for accuracy and stability reasons (see also Chapter 2.2). Then the parameters from the static solution are applied as initial conditions. This defines the frame around the dynamic analysis.

As in the static element analysis, each element must be treated separately. That means that we have to determine a stiffness-, mass- and damping matrices for each element. General stiffness matrix for a rod is found in Bergan (1989: 191)

$$\mathbf{k} = \frac{EA}{l} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.1)$$

The mass matrix takes into consideration that the mass is lumped to each node. This makes the matrix diagonal, which can be beneficial when it comes to the amount of time needed to process the dynamic analysis.

$$\mathbf{m} = \frac{ml}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.2)$$

Because we only calculate the hydrodynamic damping, it is easy to set up the damping matrix as well. By applying the damping forces to the nodes this matrix also becomes diagonal:

$$\mathbf{c} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.3)$$

Because drag forces only have an impact in the direction perpendicular to the object's surface, we get a sparse matrix. The problem is that the drag force is given on the

form $\frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot v^2$. This is equal to $\mathbf{c}\dot{\mathbf{u}}^2$. We do now not have a differential equation on the

form of eq. (2.23). Because the velocity parameter of the equation has become non linear, we need to rewrite the expression so that we have a linear equation. This can be done by the following approximation

$$\dot{\mathbf{u}}(t_n)^2 \approx \dot{\mathbf{u}}(t_n) \cdot \dot{\mathbf{u}}(t_{n-1/2}) \quad (5.4)$$

This can be further rewritten into

$$\dot{\mathbf{u}}(t_n) \cdot \dot{\mathbf{u}}(t_{n-1/2}) = \frac{u_{n+1} - u_{n-1}}{2\Delta t} \cdot \frac{u_n - u_{n-1}}{\Delta t} \quad (5.5)$$

All the variables in the second part of the right side of eq. (5.5) are known, and the first part is known from eq. (2.26). This means that our damping coefficient needs to be on the form

$$\mathbf{c} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot \frac{u_n - u_{n-1}}{\Delta t} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.6)$$

We have now established the local material matrices. But as each element has a different rotation as the next one, we need to transform them in their respective angle (Figure 5.1).

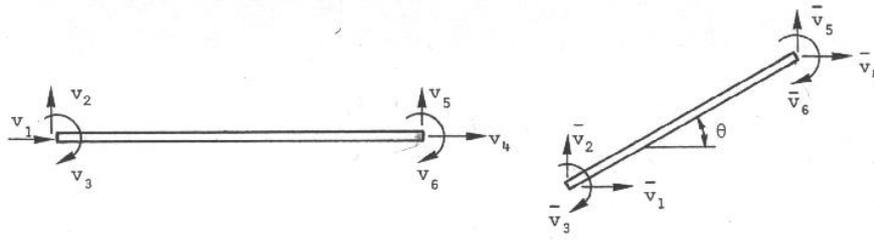


Figure 5.1: Rod element with 6 degrees of freedom.

This is done with a standard transformation matrix (Bergan 2002: 193), shown in eq. (5.7).

$$\mathbf{T} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & \cos \varphi & \sin \varphi \\ 0 & 0 & -\sin \varphi & \cos \varphi \end{pmatrix} \quad (5.7)$$

The transformation is done for each element by the following equation, here shown with the stiffness matrix as an example:

$$\bar{\mathbf{k}} = \mathbf{T}'\mathbf{k}\mathbf{T} \quad (5.8)$$

After the local matrices have been transformed to the global position for each element, they are assembled to large system matrices. These are coupled in each node and make up the matrices that will be used in the further calculations.

At this point some modifications are necessary. This includes never allowing the line to fall under the seabed, not allowing any displacement in any direction at the anchor and obstruct the end node to fall beneath the water level. As the displacement in the first time step is set to be a vector of zeros, we can now estimate the displacement at the second time step using the start up algorithm described in chapter 2.2. We now know the displacement after the second time step. This means that the line has become a different shape, and we need to update all the physical parameters such as node coordinates, angle, element length and tension. These new parameters are used to calculate new stiffness-, mass- and damping matrices. From this point on we have enough information to make up a general time loop. We have got the displacement in the first two time steps and are now able to calculate the displacement U at the time i . It is of course important to notice that the physical parameters mentioned before also needs to be updated for each time step in the general loop.

We are now able to plot the shape of the line at different time steps. We can also plot the displacement of certain nodes in any direction. By making up a routine where the damping is excluded we can plot the displacement for each node with and without damping and analyse the effects of damping.

6 Results and comments

6.1 Input parameters

The following parameters (Table 6.1) are used unless other is specified (Vryhof 2005: 162, 174). The water depth is always 300m.

Table 6.1: Input parameters.

	Chain	Fibre
Weight [kg/m]	95.5	17.2
Cross section area [m ²]	0.013616	0.0274
Elasticity modul [N/m ²]	$211 \cdot 10^5$	$5.027 \cdot 10^9$
Diameter [m]	0.157	0.187
Drag coefficient	2.4	1.8

The external load case applied in the dynamic analysis is a step force. A step force jumps suddenly from zero to p_0 and stays constant as that value (Figure 6.1). The size and duration of this load vary.

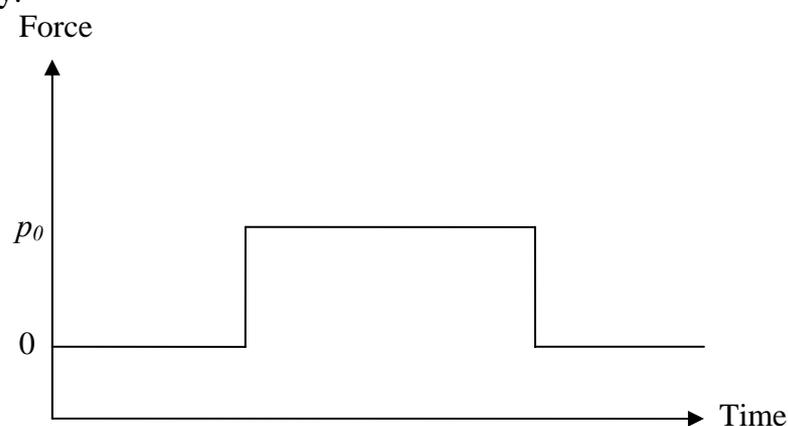


Figure 6.1: Step case.

6.2 Analytical solution

By use of MATLAB it is possible to implement the equations derived in chapter 3 and plot the shape of the line. As shown later, MATLAB will also be used to make a routine for both the static and dynamic element models. In the following a number of plots concerning the analytical static solution will be presented.

Catenary equation

First we implement eq. (3.17) to plot the shape of the line, given the geometrical boundaries calculated in Maple. Figure 6.2 shows a satisfactory relation between expected and actual form of line. This also shows that the equations referred to above gives a good approximation for the shape, and therefore is to be seen as correct applied. Figure 6.2 give us a good basis for further calculations.

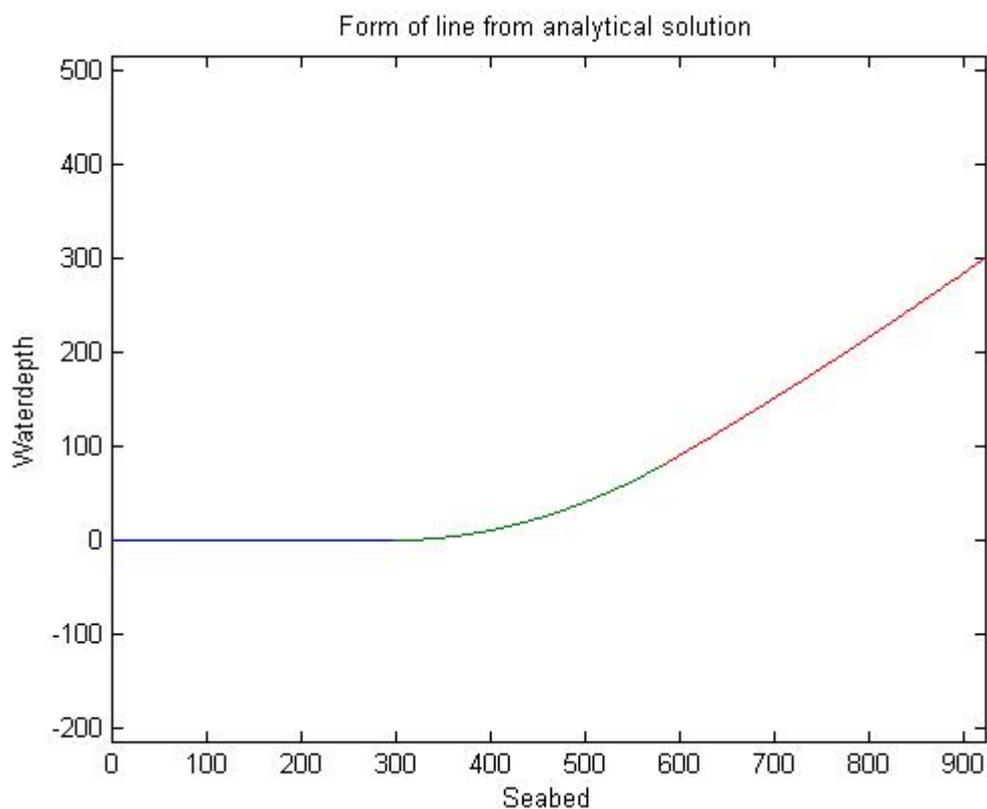


Figure 6.2: Plot of line by analytical solution.

Change of chain weight

An appropriate check that the analytical solution acts correctly is to apply a heavier material at the lower part of the line and see if the behaviour of the line is changing. In Figure 6.3 we see three cases. In each case the chain part of the line has a different weight. As expected, a bigger part of the line lies on the seabed as the weight of the chain increases. Figure 6.4 shows a more detailed part of the line to underline the behaviour described above. In Table 6.2 the values of different parameters are shown. These are read directly from the solution-file in MATLAB, and give us a more defined picture of the difference in each case. The dotted line represents the case with the lowest weight.

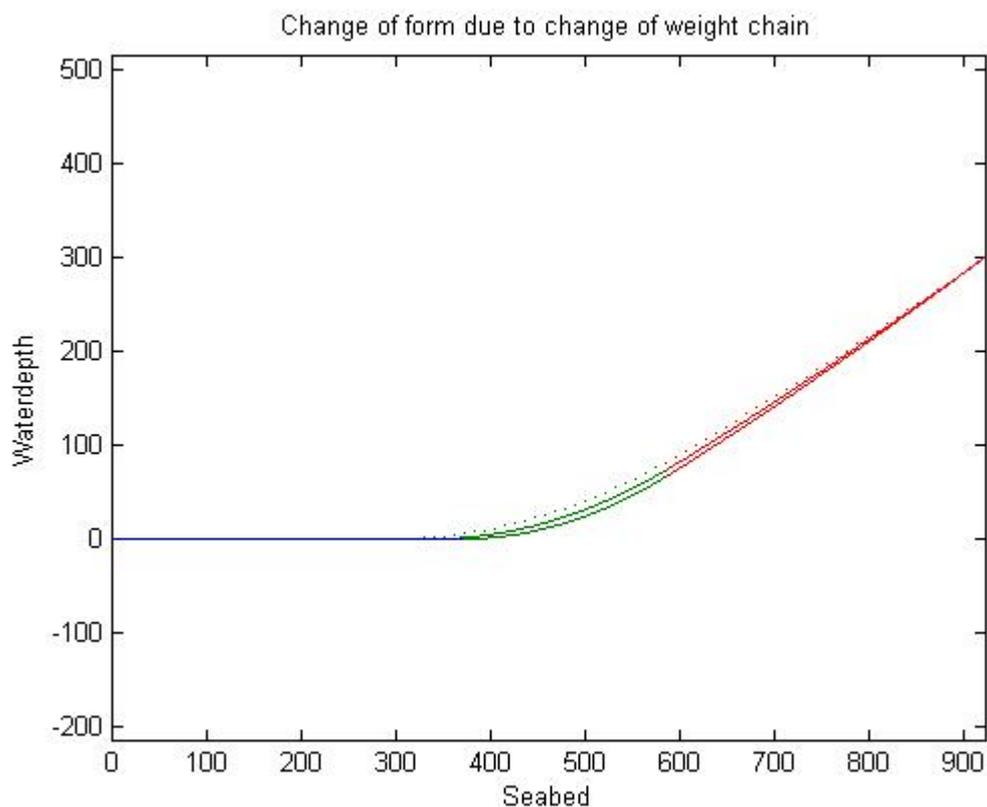


Figure 6.3: Plot of line with different weights in chain part.

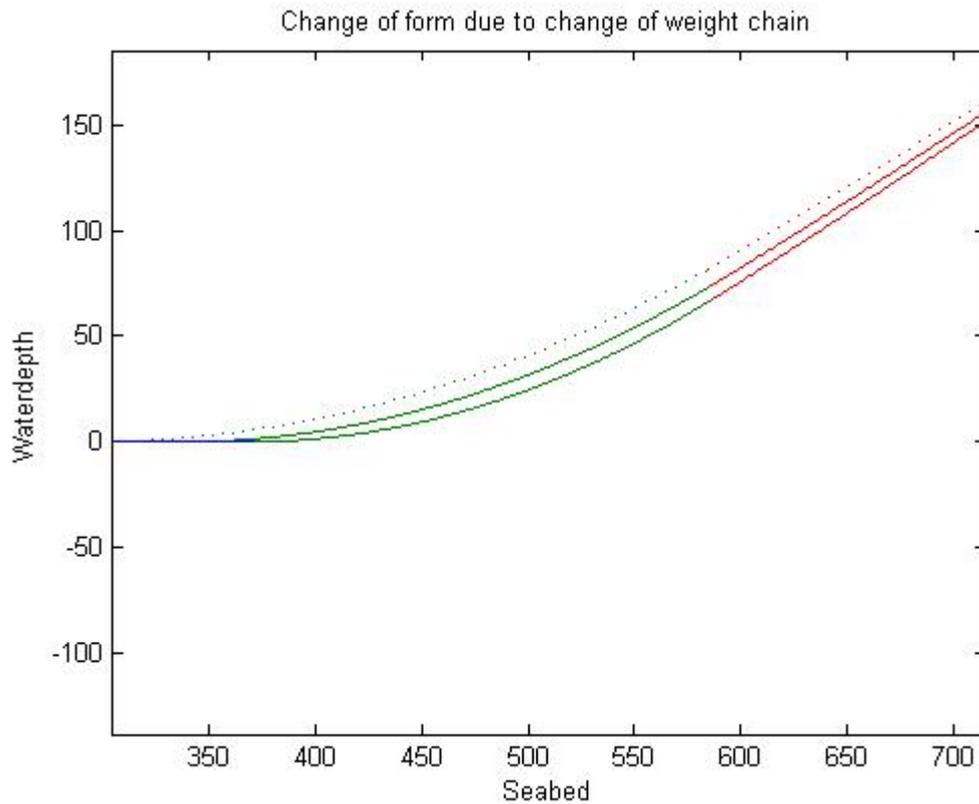


Figure 6.4: Detailed plot of lower part of anchor line.

Table 6.2: Parameters at different chain weight cases.

	$w_l = 936.855$ [kg/m]	$w_l = 1133.055$ [kg/m]	$w_l = 1329.255$ [kg/m]
l_l [m]	302.9	260.1	228.5
h_l [m]	81.7	73.7	67.3
x_l [m]	287.7	246.0	215.1
l_b [m]	297.4	339.9	371.5
G_x [N]	$4.9 \cdot 10^5$	$4.8 \cdot 10^5$	$4.7 \cdot 10^5$
G_y [N]	$3.5 \cdot 10^5$	$3.6 \cdot 10^5$	$3.7 \cdot 10^5$
φ_w	35.8°	37.1°	38.2°

In Table 6.2 the parameter x_l refer to the distance between the point where the line leaves the seabed and the point where the line changes material. See also Figure 3.1 for further information.

Change of initial tension

By increasing the tension in the line one would expect that the line would get a decrease in the angle at the water plane, and that a shorter part of the chain would lie on the seabed (Figure 6.5). This is also a good check that the analytical solution is correct. Figure 6.6 shows a detailed part of the line, and Table 6.3 gives us the exact values of the different parameters. The dotted line represents the case with the lowest tension.

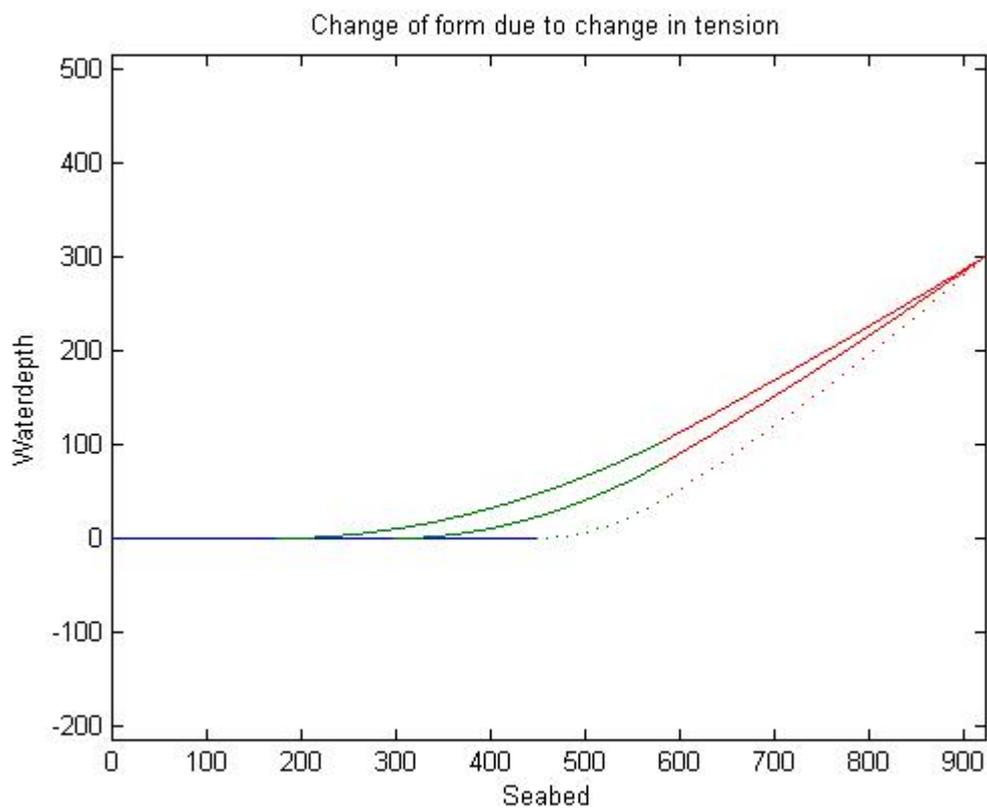


Figure 6.5: Plot of line with different initial tension.

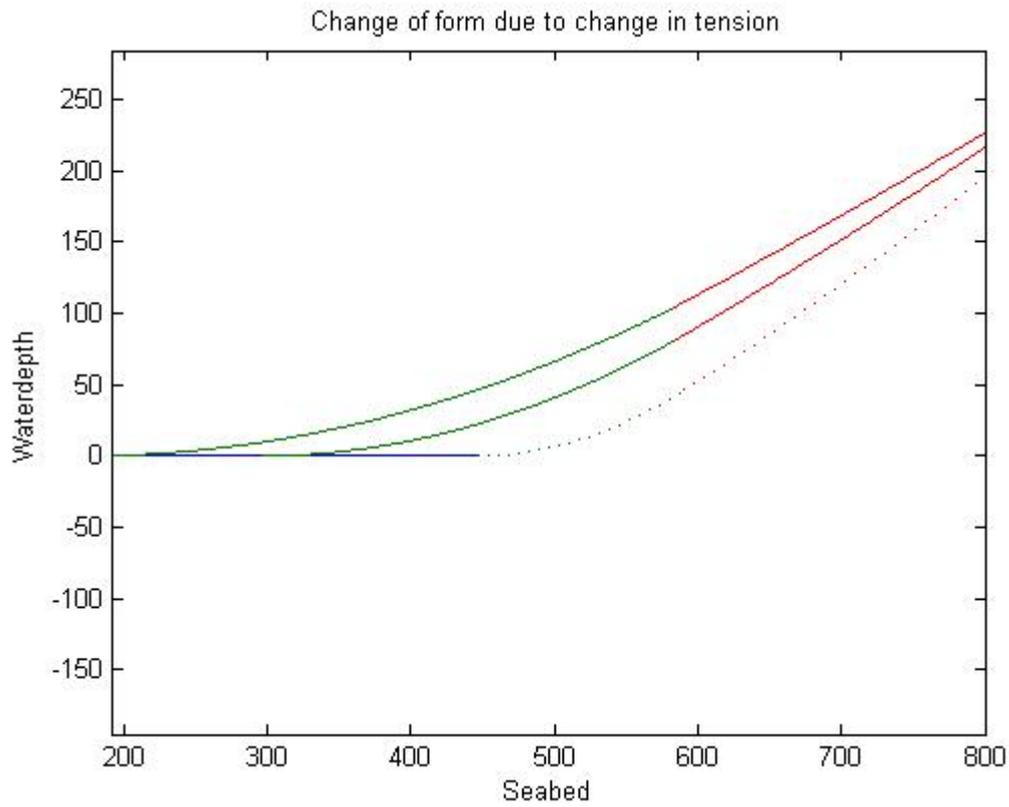


Figure 6.6: Detailed part of line with different initial tension.

Table 6.3: Parameters at different initial tension cases.

	$T = 3 \cdot 10^5$ [N]	$T = 6 \cdot 10^5$ [N]	$T = 9 \cdot 10^5$ [N]
l_l [m]	152.5	302.6	426.0
h_l [m]	46.3	81.7	103.9
x_l [m]	143.0	287.7	409.0
l_b [m]	447.5	297.4	174.0
G_x [N]	$2.1 \cdot 10^5$	$4.9 \cdot 10^5$	$7.7 \cdot 10^5$
G_y [N]	$2.1 \cdot 10^5$	$3.5 \cdot 10^5$	$4.7 \cdot 10^5$
φ_w	44.5°	35.8°	31.2°

6.3 Comparison of analytical model and static element model

The interesting part is to see if the analytical solution is more or less equal to the element solution. If they are, then we have succeeded in obtaining a good approximation for the initial conditions we need in order to perform a dynamic analysis. In Figure 6.7 and 6.8 we see that the two solutions are close to identical. From Table 6.4 we see that the important point where the two materials meet is concurrent.

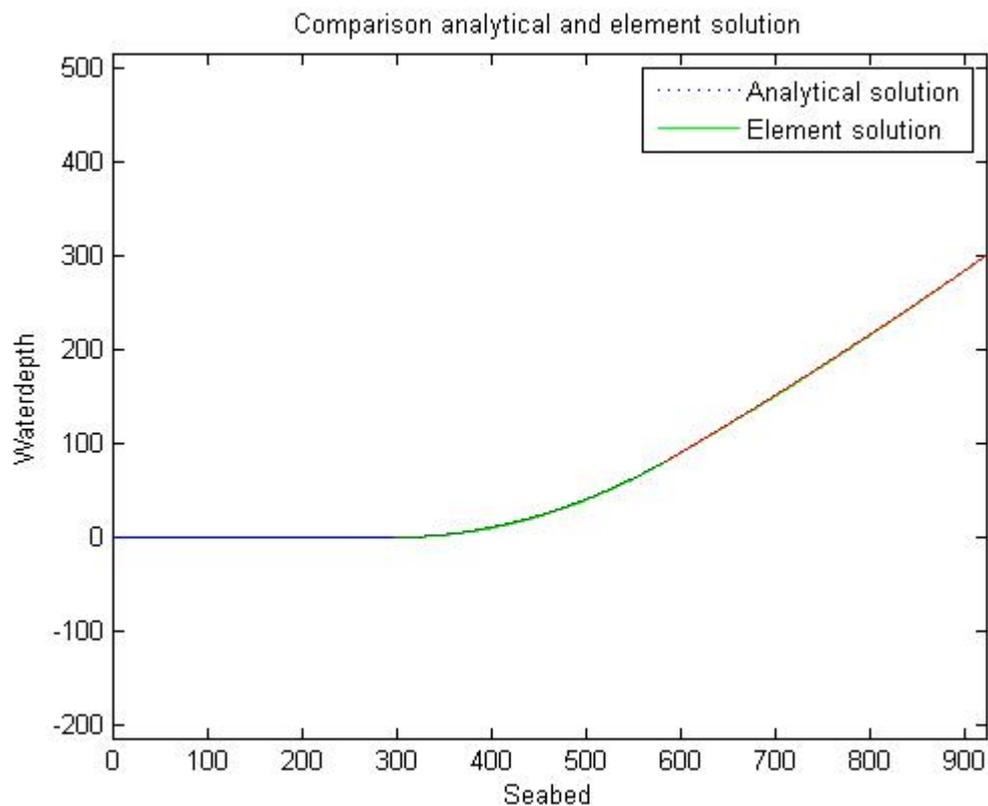


Figure 6.7: Comparison analytical and element solution.

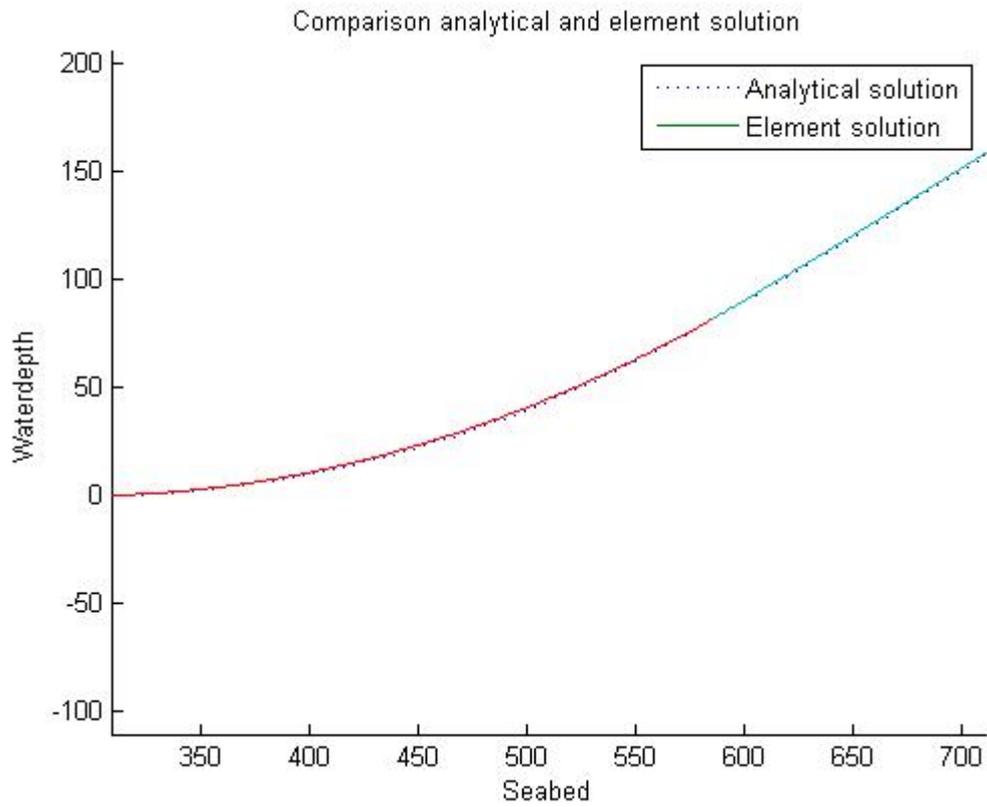


Figure 6.8: Detailed part of comparison.

Table 6.4: Comparison of parameters from analytical and element solution.

	l1	h1	x1
Analytical solution	302.6	81.7	287.7
Element solution	302	81.0	286.3

6.4 Dynamic model without damping

A load of size 4000 kN is applied as a step force over a time period of 8 seconds. In MATLAB this force is applied from the very first time step. As expected of an undamped system, the line experiences large oscillations within short amount of time (Figure 6.9). This causes large displacement in the node-coordinates (Figure 6.10 and 6.11). It also causes the line to lift from the seabed, which is an unwanted situation. Due to computational work time the line only consists of a total of 9 elements. By increasing the number of elements the plot of the line will be smoother, but few elements still give a good approximation of line behaviour.

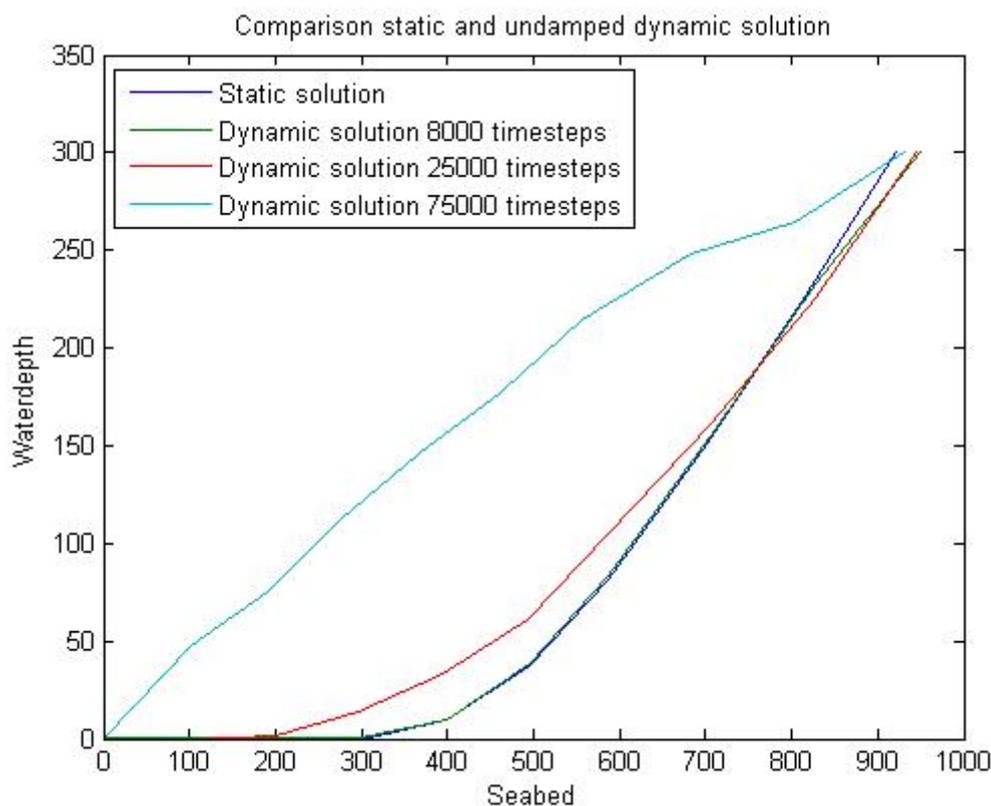


Figure 6.9: Undamped line at various time steps.

During the first second the load is applied the node at the end of the line makes a jump of 33 meters in horizontal direction. The end-node is denied any vertical displacement to prevent it from rising above or sinking beneath sea level. This means that the end-node does not have a vertical displacement, so to get a plot of this parameter plot the displacement in the node number two from the top. This gives us a displacement of nearly 10 meters the second after the load is applied.

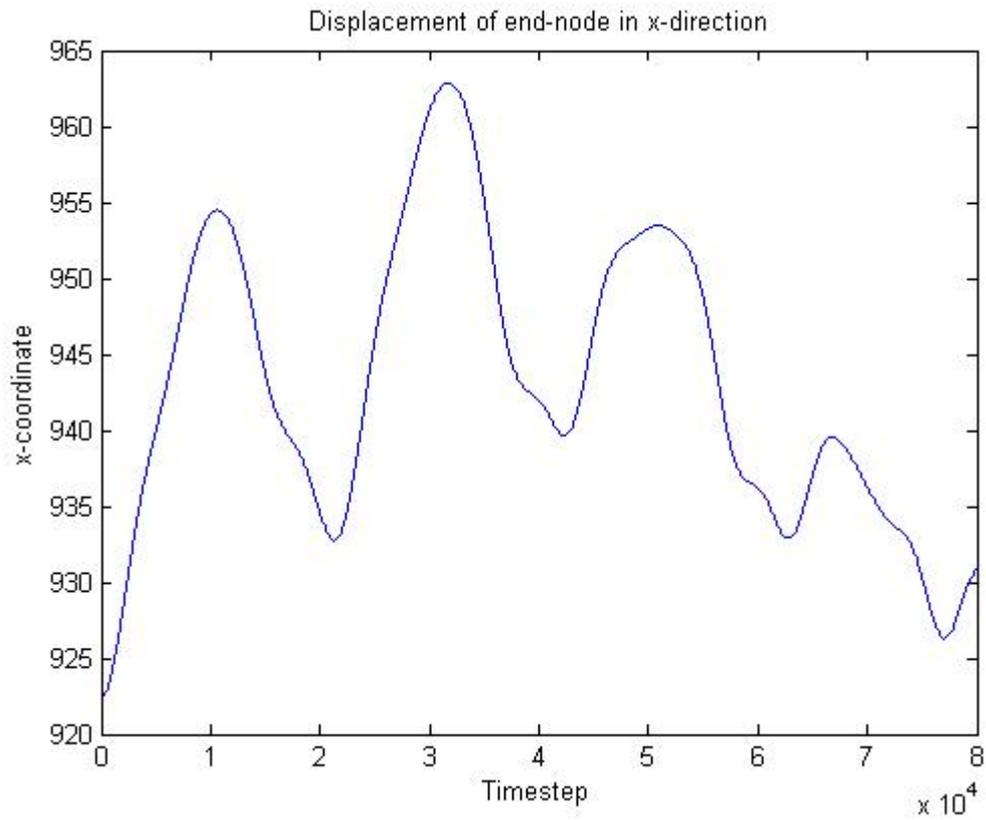


Figure 6.10: Undamped displacement in x-direction.

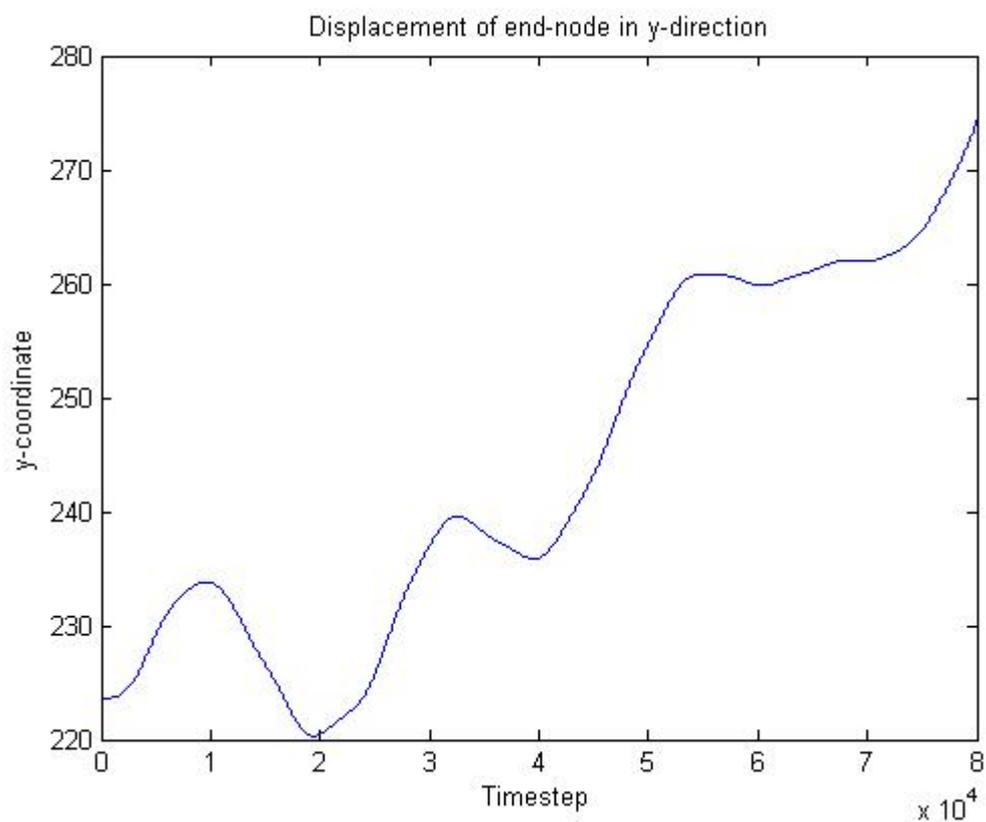


Figure 6.11: Undamped displacement in y-direction.

6.5 Dynamic model with damping

When damping is included in the system the vibrations steadily diminishes in amplitude. This means that the oscillations experienced earlier will be smaller (Figure 6.12). The behaviour of the line is steadier and the deformations are minor. The total of the line is still rising above the seabed, which is undesirable. This problem is also restricted to a total of 9 elements.

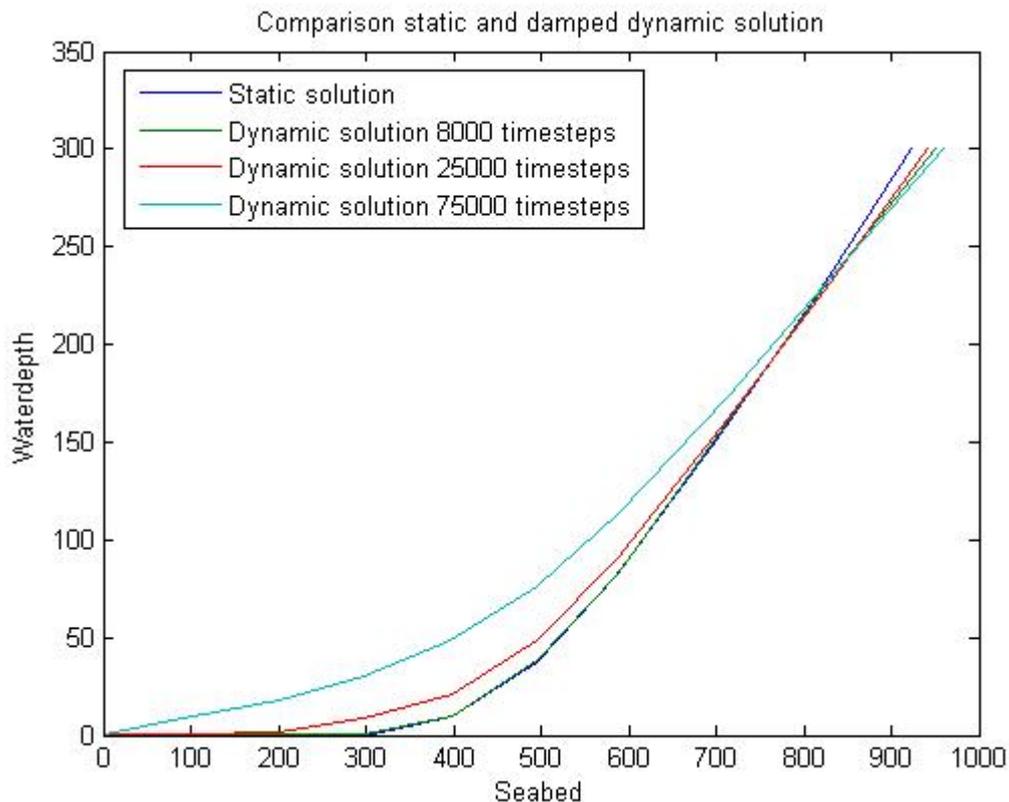


Figure 6.12: Damped line at various time steps.

For comparison the undamped displacement is implemented in Figure 6.13 and 6.14 as dotted lines. In both horizontal and vertical direction the displacements have decreasing amplitudes. The behaviour is also steadier and more controlled.

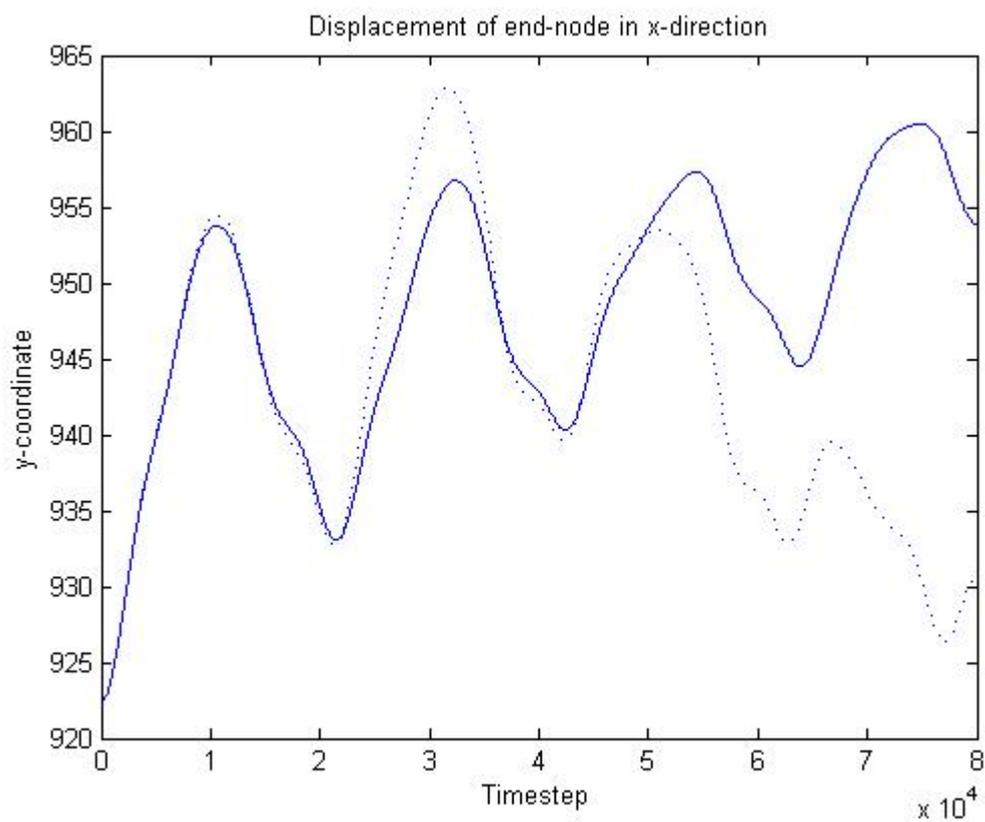


Figure 6.13: Damped displacement in x-direction.

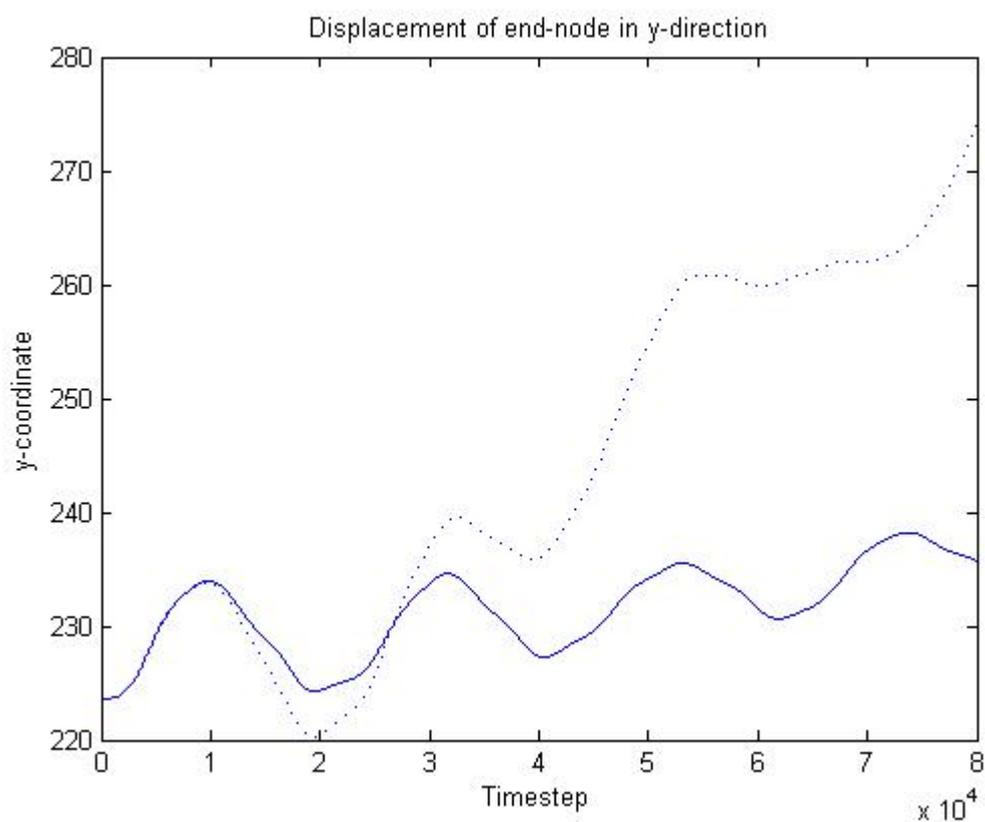


Figure 6.14: Damped displacement in y-direction.

6.6 SIMO model

In SIMO a step load of 100 kN is applied at the 200th second. Damping is included in the calculations. SIMO uses a “shooting method” to estimate the mooring line configuration. This is iteration on the boundary conditions at one end in order to satisfy specified conditions at the other. SIMO calculates the problem with two lines instead of one, and also applies the body holding the lines together. This means that SIMO takes into consideration the inertia forces from the windmill. The windmill gets a displacement of almost 60 meters at the instant the load is applied (Figure 6.15). It then bounces back to its starting position and oscillates around that point for the rest of the time. The amplitudes of the oscillations decrease, as one would expect in a damped system.

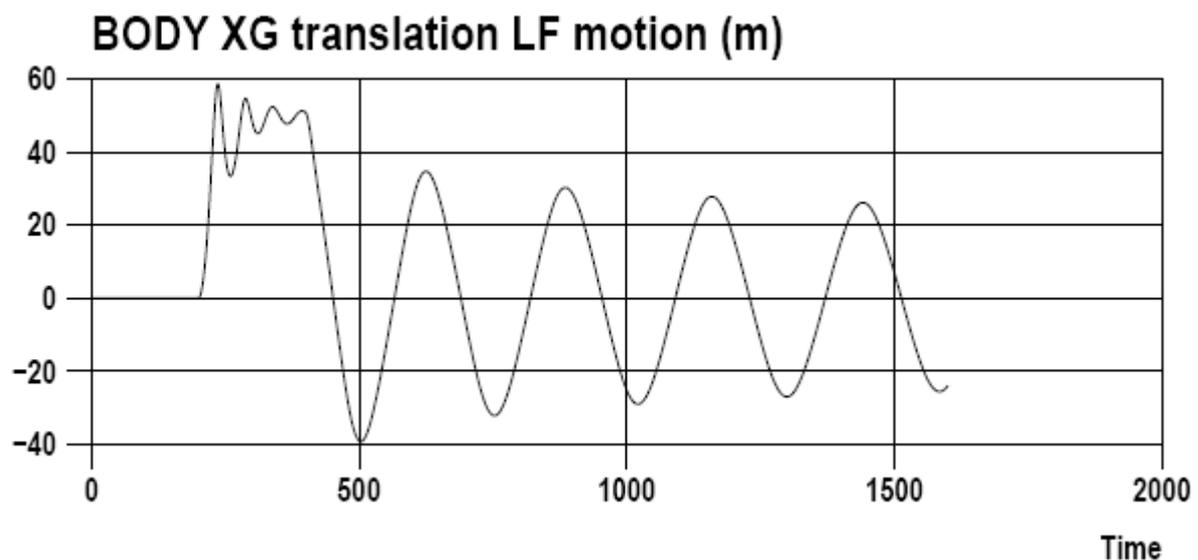


Figure 6.15: Displacement in x-direction from SIMO.

In vertical direction the windmill drops about 3.5 meters under the surface when the load case is applied (Figure 6.16). The displacement is smaller than in horizontal direction, and the oscillation is damped out fairly quick. As the windmill is a larger body with corresponding buoyancy, this is as expected.

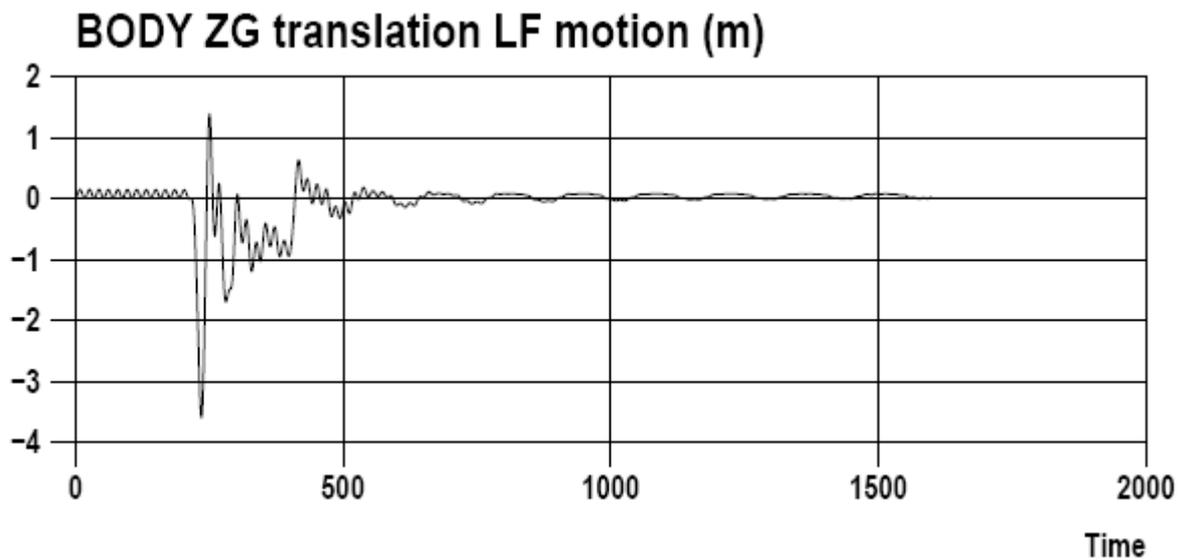


Figure 6.16: Displacement in y-direction from SIMO.

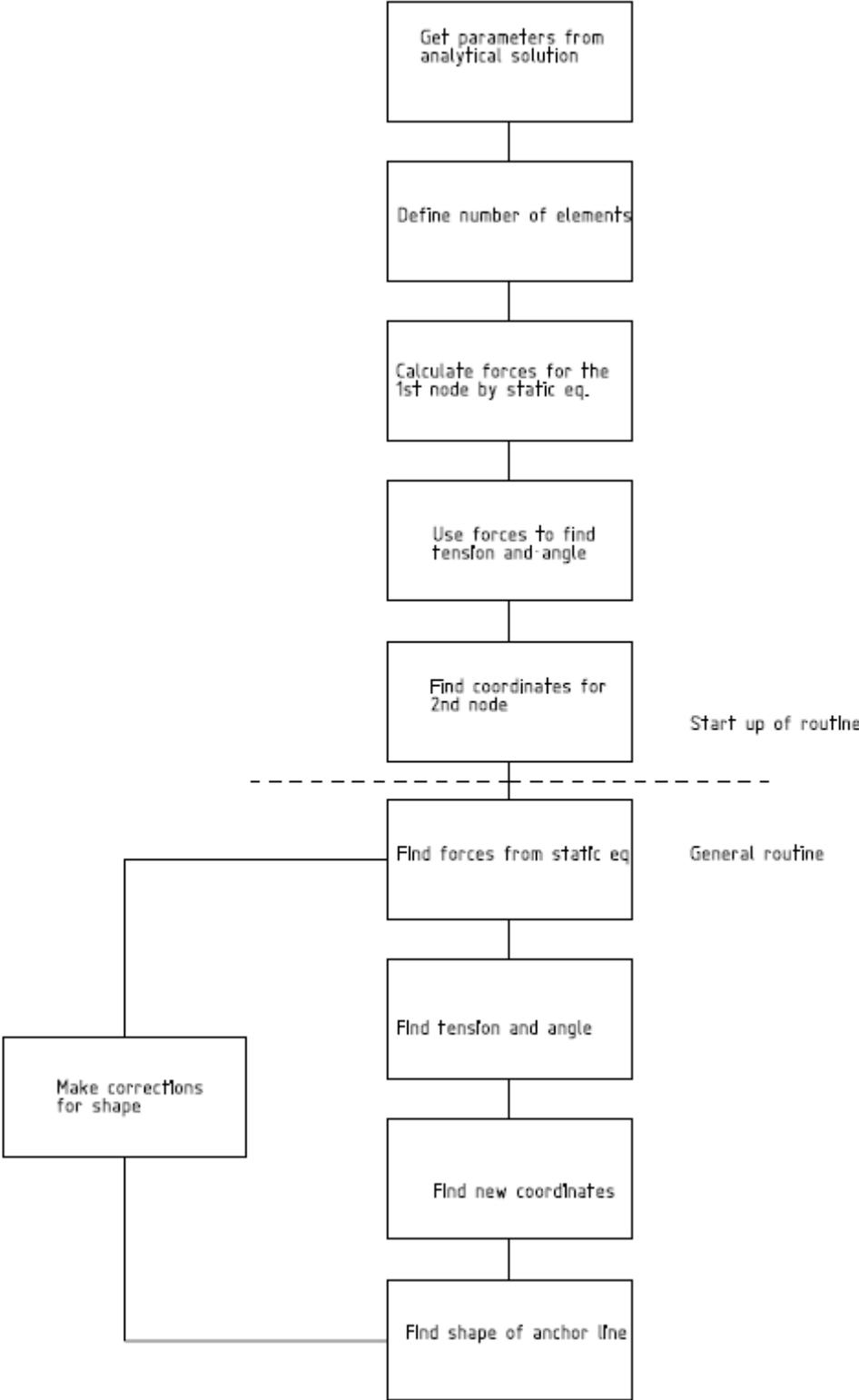
7 Conclusion

In the numerical solution computed in MATLAB, one has only modelled the line from its anchor point up to the water level. In order to keep the end point at the water level, the line is restricted any vertical displacement at the top in a relatively brutal manner. SIMO computes an entire system including two lines and a body holding the lines together. In SIMO the mass forces on lines are neglected, and in MATLAB the inertia forces from the body/windmill is neglected. In addition, due to computation time in MATLAB, the numerical solution is modelled with few elements and over a short period of time. All of the reasons listed above may be decisive to the leap in results, with the biggest leap being the difference in horizontal displacement for the damped systems. As it stands, SIMO seems like the best option for solving a dynamic problem over a longer period of time, and MATLAB seems like the better option for a more detailed picture of a shorter period of time. The difference in results may come from one or more of the alternatives above.

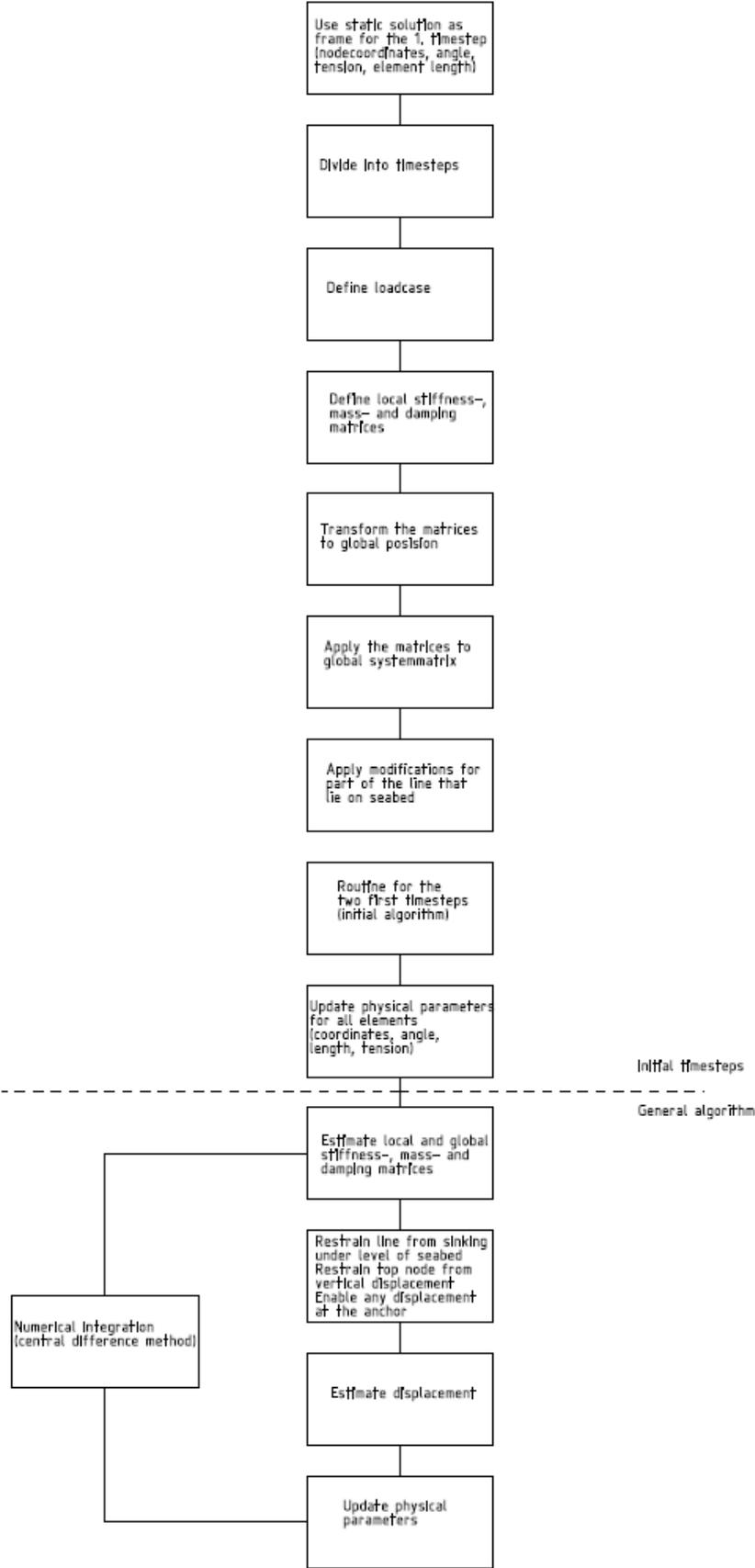
References

- Bergan, Pål G. and Tor G. Syvertsen. 1989. *Knekning av søyler og rammer*. 2nd ed. Trondheim: Tapir forlag.
- Calvert, J. B. 2000. *The Catenary*. [Online]. Colorado: University of Denver. Available from URL: < <http://mysite.du.edu/~jcalvert/math/catenary.htm>>. [Accessed 3.may 2008].
- Chopra, Anil K. 2007. *Dynamics of Structures*. 3rd ed. Upper Saddle River, NJ: Pearson Education, Inc.
- Faltinsen, O. M. 1990. *Sea loads on ships and offshore structures*. New York, NY: Press Syndicate of the University of Cambridge.
- Irgens, Fridtjov. 1992. *Formelsamling mekanikk*. 2nd ed. Trondheim: Tapir forlag.
- Osnes, Harald. 2007. *Kap. 5: Numeriske løsningsmetoder* [Online]. Documents presented in course MEK 4510 - Svingninger i konstruksjoner. Available from URL: < http://www.math.uio.no/~osnes/kurs/MEK4510/www/mek4510_kap5_4p_07.pdf>. [Accessed 2007 February 19].
- Pearson, Nicholas John. 2002. Experimental Snap Loading of Synthetic Fiber Rope. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg.
- SIMO project team. 2004. *SIMO – User's Manual Version 3.4*. Trondheim: SINTEF.
- Swicofil. 2008. *Recreation and Leisure*. Switzerland: Emmenbruecke. Available from URL: < <http://www.swicofil.com/vectran.html>>. [Accessed 2008 March 15].
- Vryhof Anchors. 2005. *Vryhof Anchor Manual 2005*. The Netherlands: Krimpen ad Yssel. Available from URL: < <http://www.vryhof.com/PDF/AM2000.pdf>>. [Accessed 2008 February 3].
- Wakecord. 2007. *Rope And Related Uses For Dyneema® and Spectra®*. The US. Available from URL: < <http://www.wakecord.com/spectra.htm>>. [Accessed 2008 March 15].

Appendix A: Flow diagram of static element solution



Appendix B: Flow diagram of dynamic element solution



Appendix C: Calculations made in Maple

```

> l2 := 400
                                l2 := 400
                                (1)
> lt := l1 + l2
                                lt := l1 + 400
                                (2)
> w1 := 135.5·9.81
                                w1 := 1329.255
                                (3)
> w2 := 17.2·9.81
                                w2 := 168.732
                                (4)
> ht := 300
                                ht := 300
                                (5)
> h2 := ht - h1
                                h2 := 300 - h1
                                (6)
> A1 := 0.013616
                                A1 := 0.013616
                                (7)
> E1 := 211·109
                                E1 := 211000000000
                                (8)
> d2 := 187
                                d2 := 187
                                (9)
> A2 :=  $\frac{3.14 \cdot \left(\frac{d2}{2}\right)^2}{10^6}$ 
                                A2 := 0.02745066500
                                (10)
> E2 := 5.0272·109
                                E2 := 5.027200000 109
                                (11)
> T := 600000
                                T := 600000
                                (12)
>
> lt := l1 + l2
                                lt := l1 + 400
                                (13)
> Gy := w1·l1 + w2·l2
                                Gy := 1329.255 l1 + 67492.800
                                (14)
>
> Gx1 :=  $\sqrt{T^2 - Gy^2}$ 
                                Gx1 :=  $\sqrt{360000000000 - (1329.255 l1 + 67492.800)^2}$ 
                                (15)
> Fy := w1·l1
                                Fy := 1329.255 l1
                                (16)

```

$$\begin{aligned}
 > a := w1 \cdot h1 - \frac{1}{2} \cdot \frac{w1^2 \cdot l1^2}{A1 \cdot EI} \\
 & \qquad \qquad \qquad a := 1329.255 \, h1 - 0.0003075067204 \, l1^2
 \end{aligned} \tag{19}$$

$$\begin{aligned}
 > Fx := \frac{(Fy^2 - a^2)}{2 \cdot a} \\
 Fx := \frac{1}{2} \frac{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2}{1329.255 \, h1 - 0.0003075067204 \, l1^2}
 \end{aligned} \tag{20}$$

$$\begin{aligned}
 > b := \frac{w2 \cdot h2}{Fx} - \frac{1}{2} \cdot \frac{w2^2 \cdot l2^2}{A2 \cdot E2 \cdot Fx} + \frac{\sqrt{Fx^2 + Fy^2}}{Fx} \\
 b := \frac{337.464 (300 - h1) (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} \\
 - \frac{33.00926528 (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2}
 \end{aligned} \tag{21}$$

$$\left. \left(\frac{337.464 (300 - h1) (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} - \frac{33.00926528 (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} \right) \right/ \left(1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2 \right)$$

$$\begin{aligned}
 > Gx2 := \frac{Gy}{\sqrt{b^2 - 1}} \\
 Gx2 := (1329.255 \, l1 + 67492.800) / \left(\left(\frac{337.464 (300 - h1) (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} - \frac{33.00926528 (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} \right)^2 - 1 \right)
 \end{aligned} \tag{22}$$

$$\left(\left(\frac{337.464 (300 - h1) (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} - \frac{33.00926528 (1329.255 \, h1 - 0.0003075067204 \, l1^2)}{1.766918855 \, 10^6 \, l1^2 - (1329.255 \, h1 - 0.0003075067204 \, l1^2)^2} \right)^2 - 1 \right)$$

1/2

> $Likninger := \{Fx = Gx2, Gx1 = Gx2\}$

$$Likninger := \left\{ \frac{1}{2} \frac{1.766918855 \cdot 10^6 l^2 - (1329.255 h l - 0.0003075067204 l^2)^2}{1329.255 h l - 0.0003075067204 l^2} \right. \quad (23)$$

$$= (1329.255 l l + 67492.800) /$$

$$\left(\left(\frac{337.464 (300 - h l) (1329.255 h l - 0.0003075067204 l^2)}{1.766918855 \cdot 10^6 l^2 - (1329.255 h l - 0.0003075067204 l^2)^2} \right. \right.$$

$$\left. \left. - \frac{33.00926528 (1329.255 h l - 0.0003075067204 l^2)}{1.766918855 \cdot 10^6 l^2 - (1329.255 h l - 0.0003075067204 l^2)^2} \right) \right)$$

Appendix D: MATLAB-code for dynamic solution with damping

```
function [] = dynamisk__med_demping()

minL1= 100;
maxL1 =500;

for j = minL1:1:maxL1;
[x,y] = statisk(j,3,3,3);
%[x,y] = statisk(j,2,2,2);
h(j) = max(y);
end

error = (300 - h).^2;

% Llguess = 1:maxL1;
% figure(1)
% plot(Llguess,h)
% xlabel('Guess of L1')
% ylabel('Corresponding waterdepth')
% figure(2)
% plot(Llguess,error)
% xlabel('Guess of L1')
% ylabel('Error')

[a,L1]=min(error);
[x,y,Fx,Fy,phi,w,elm_length,A,E,no_elms_total,no_elms_chain,no_elms_fiber,no_elms_bottom, T,w_chain,w_fiber] = statisk(L1,3,3,3);
%[x,y,Fx,Fy,phi,w,elm_length,A,E,no_elms_total,no_elms_chain,no_elms_fiber,no_elms_bottom, T,w_chain,w_fiber] = statisk(L1,2,2,2);

dynamisk_demp(x,y,Fx,Fy,phi,w,elm_length,A,E,no_elms_total,no_elms_chain,no_elms_fiber,no_elms_bottom,T,w_chain,w_fiber);

%-----
% Definisjon av funksjoner
%-----
function
[x,y,Fx,Fy,phi,w,elm_length,A,E,no_elms_total,no_elms_chain,no_elms_fiber,no_elms_bottom, T,w_chain,w_fiber] = statisk(length_chain_above, no_elms_bottom, no_elms_above, no_elms_fiber)

no_elms_chain = no_elms_bottom + no_elms_above;
no_elms_total = no_elms_chain + no_elms_fiber;

length_chain = 600;
length_chain_bottom = length_chain - length_chain_above;

length_elm_bottom = length_chain_bottom/no_elms_bottom;
length_elm_above = length_chain_above/no_elms_above;

%w_chain = 936;
w_chain = 0.000936;
%E_chain = 211000000000;
E_chain = 211000.0;
A_chain = 0.013616;

length_fiber = 400;
length_elm_fiber = length_fiber/no_elms_fiber;

% w_fiber = 168.732;
w_fiber = 0.000168732;
% E_fiber = 5.027200000*10^9;
E_fiber = 5.0272*10^3;
```

```

A_fiber = 0.02745066500;

% Coordinates of the nodal points when axial strains are neglected
x0 = zeros((no_elms_total+1),(1));
y0 = zeros((no_elms_total+1),(1));

% Coordinates of the nodal points when the effect of axial strains are included
x = zeros((no_elms_total+1),(1));
y = zeros((no_elms_total+1),(1));

Fx = zeros((no_elms_total),(1));
Fy = zeros((no_elms_total),(1));
T = zeros((no_elms_total),(1));
phi = zeros((no_elms_total),(1));

w = zeros((no_elms_total),(1));

for i = 1:no_elms_chain, w(i) = w_chain; end
for i = no_elms_chain+1:no_elms_total, w(i) = w_fiber; end

elm_length = zeros((no_elms_total),(1));
for i = 1:no_elms_bottom, elm_length(i) = length_elm_bottom; end
for i = no_elms_bottom+1:no_elms_chain, elm_length(i) = length_elm_above; end
for i = no_elms_chain+1:no_elms_total, elm_length(i) = length_elm_fiber; end

E = zeros((no_elms_total),(1));
for i = 1:no_elms_chain, E(i) = E_chain; end
for i = no_elms_chain+1:no_elms_total, E(i) = E_fiber; end

A = zeros((no_elms_total),(1));
for i = 1:no_elms_chain, A(i) = A_chain; end
for i = no_elms_chain+1:no_elms_total, A(i) = A_fiber; end

%Tmax=600000;
Tmax = 0.6;

% Spesialbehandling av element 1 (og knutepunkt 2)

% Krefter fra element "0"
Fx0 = sqrt(Tmax^2 - (w_chain*length_chain_above + w_fiber*length_fiber)^2);
Fy0 = 0.0;

% Kraftlikevekt for knutepunkt 1 gir:
Fx(1) = Fx0;
Fy(1) = Fy0;

% Total kraft for element 1 blir da:
T(1) = sqrt(Fx(1)^2 + Fy(1)^2);

phi(1) = atan(Fy(1)/Fx(1));

% Koordinatene til knutepunkt 2 blir:
x0(2) = x0(1) + elm_length(1)*cos(phi(1));
y0(2) = y0(1) + elm_length(1)*sin(phi(1));

x(2) = elm_length(1)*(1.0 + T(1)/(A(1)*E(1)))*cos(phi(1));
y(2) = elm_length(1)*(1.0 + T(1)/(A(1)*E(1)))*sin(phi(i));

% Resten av elementene som ligger langs bunnen
for i = 2:(no_elms_bottom)
    Fx(i) = Fx(i-1);
    Fy(i) = Fy(i-1);
    T(i) = sqrt(Fx(i)^2 + Fy(i)^2);

    x0(i+1) = x0(i) + elm_length(i)*cos(phi(i));

```

```

    y0(i+1) = y0(i) + elm_length(i)*sin(phi(i));

    x(i+1) = x(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*cos(phi(i));
    y(i+1) = y(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*sin(phi(i));
end

% Spesialbehandling av knutepunktet som grenser til det første elementet som
forlater grunnen

% Ingen løkke, men en bestemt verdi for i
i = no_elms_bottom+1;
    Fx(i) = Fx(i-1);
    Fy(i) = Fy(i-1) + 0.5*w(i)*elm_length(i);
    T(i) = sqrt(Fx(i)^2 + Fy(i)^2);
    phi(i) = atan(Fy(i)/Fx(i));

    x0(i+1) = x0(i) + elm_length(i)*cos(phi(i));
    y0(i+1) = y0(i) + elm_length(i)*sin(phi(i));

    x(i+1) = x(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*cos(phi(i));
    y(i+1) = y(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*sin(phi(i));

% Løkke for resten av elementene
for i = no_elms_bottom+2:no_elms_total;
    Fx(i) = Fx(i-1);
    Fy(i) = Fy(i-1) + 0.5*(w(i-1)*elm_length(i-1) + w(i)*elm_length(i));
    T(i) = sqrt(Fx(i)^2 + Fy(i)^2);
    phi(i) = atan(Fy(i)/Fx(i));

    x0(i+1) = x0(i) + elm_length(i)*cos(phi(i));
    y0(i+1) = y0(i) + elm_length(i)*sin(phi(i));

    x(i+1) = x(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*cos(phi(i));
    y(i+1) = y(i) + elm_length(i)*(1.0 + T(i)/(A(i)*E(i)))*sin(phi(i));
end
%-----
function [U] =
dynamisk_demp(x,y,Fx,Fy,phi,w,elm_length,A,E,no_elms_total,no_elms_chain,no_elms_fi
ber,no_elms_bottom,T,w_chain,w_fiber)
%Dynamisk del av analyse

% Dele opp i tidsskritt
delta_t = 0.0001;          %Tidsskritt
tid = 8.0;                 %antall sekunder
ant_tidsskritt = tid/delta_t;

% Definere str på matrisene
matrix_size = 2*no_elms_total + 2;

% ----Legge inn den statiske løsningen som ramme i første tidsskritt----
% Nye nodekoordinater og ny vinkel for hvert element
x_new = zeros(no_elms_total + 1,ant_tidsskritt);
x_new(:,1) = x;
y_new = zeros(no_elms_total + 1,ant_tidsskritt);
y_new(:,1) = y;
phi_new = zeros(no_elms_total,ant_tidsskritt);
phi_new(:,1) = phi;

% Nye strekk-/kraftverdier for hvert element
Fx_new = zeros(no_elms_total,ant_tidsskritt);
Fx_new(:,1) = Fx;
Fy_new = zeros(no_elms_total,ant_tidsskritt);
Fy_new(:,1) = Fy;
strek = zeros(no_elms_total,ant_tidsskritt);
strek(:,1) = T;

% Ny lengde for hvert element
elm_length_new = zeros(no_elms_total,ant_tidsskritt);

```

```

    elm_length_new(:,1) = elm_length;
%-----
%rho_vann = 1025;
rho_vann = 0.001025;
Cd_chain = 2.4;
Cd_fiber = 1.8;

D_chain = 0.157;
D_fiber = 0.187;

Cd = zeros((no_elms_total),(1));
for j = 1:no_elms_chain, Cd(j) = Cd_chain; end
for j = no_elms_chain+1:no_elms_total, Cd(j) = Cd_fiber; end

D = zeros((no_elms_total),(1));
for j = 1:no_elms_chain, D(j) = D_chain; end
for j = no_elms_chain+1:no_elms_total, D(j) = D_fiber; end

% Definerer vektor med laster (antar samme last i hvert tidsskritt)
P = zeros(matrix_size,ant_tidsskritt);
P(matrix_size - 1,:) = 4.0;

% Sette sammen stivhets- og massematrise
K_total = zeros(matrix_size,matrix_size);
M_total = zeros(matrix_size,matrix_size);
C_total = zeros(matrix_size,matrix_size);

% Definere stivhetsmatrise og massematrise for hvert element
for e = 1:no_elms_total
    c = cos(phi_new(e,1));
    s = sin(phi_new(e,1));
    transf_matrix = [c s 0 0; -s c 0 0; 0 0 c s; 0 0 -s c];

    k_loc = (E(e)*A(e))/(elm_length_new(e,1))*[1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0
0];
    k_glob = transf_matrix' * k_loc * transf_matrix;

    m_loc = (w(e)*elm_length_new(e,1)/2)*[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
    m_glob = transf_matrix' * m_loc * transf_matrix;

%    c_loc = 1/2 * rho_vann * Cd(e) * D(e) * nonlinear(e,1) * elm_length_new(e,1) *
[1 0 0 0; 0 0 0 0; 0 0 1 0; 0 0 0 0];
%    c_glob = transf_matrix' * c_loc * transf_matrix;

    for i = 1:4
        iglob = (e-1)*2 + i;

        for j = 1:4
            jglob = (e-1)*2+ j;

            K_total(iglob, jglob) = K_total(iglob, jglob) + k_glob(i,j);
            M_total(iglob, jglob) = M_total(iglob, jglob) + m_glob(i,j);

% Med vår lineariserte versjon av dempningen blir dempningsmatrisen for 1.tidssteg
= 0.
%            C_total(iglob, jglob) = C_total(iglob, jglob) + c_glob(i,j);
        end
    end
end

% Definere str på forskyvningsmatrise
U = zeros(matrix_size,ant_tidsskritt);

P_eff = P(:,1)*0.5*(delta_t^2);

% Fordi deler av kjettingen ligger langs bunnen blir massematrisen
% singular. Den må derfor modifieres. For å gjøre det på en generell

```

```

% måte innfører vi en variabel som holder styr på antall elementer
% som ligger langs bunnen.

M_total(1,:) = 0.0;
M_total(:,1) = 0.0;
M_total(1,1) = 1;
P_eff(1) = 0.0;

for i = 1:(no_elms_bottom+1)
    index = (i-1)*2 + 2;
    M_total(index,:) = 0.0;
    M_total(:,index) = 0.0;
    M_total(index,index) = 1;
    P_eff(index) = 0;
end

M_total(matrix_size,:) = 0.0;
M_total(:,matrix_size) = 0.0;
M_total(matrix_size,matrix_size) = 1.0;
P_eff(matrix_size) = 0.0;

%U(:,2) = M_total\P_eff;
for e = 1:matrix_size;
    U(e,2) = P_eff(e)/M_total(e,e);
end

% Oppdatere fysiske parametre
for e = 1:no_elms_total+1;
    % Endring av koordinater og vinkel i hver node
    x_new(e,2) = U(2*e-1,2) + x_new(e,1);
    y_new(e,2) = U(2*e,2) + y_new(e,1);
end

for e = 1:no_elms_total;
    phi_new(e,2) = atan((y_new(e+1,2)-y_new(e,2))/(x_new(e+1,2)-x_new(e,2)));

    elm_length_new(e,2) = sqrt((x_new(e+1,2)-x_new(e,2))^2 + (y_new(e+1,2)-
y_new(e,2))^2);

    strekk(e,2) = strekk(e,1) + E(e)*A(e)*(elm_length_new(e,2)-
elm_length_new(e,1))/elm_length_new(e,1);

    Fx_new(e,2) = strekk(e,2)*cos(phi_new(e,2));
    Fy_new(e,2) = strekk(e,2)*sin(phi_new(e,2));
end
%-----Generell tidsløkke-----
for i = 3:ant_tidsskritt

    M_total(:, :) = 0.0;
    K_total(:, :) = 0.0;
    C_total(:, :) = 0.0;

    for e = 1:no_elms_total
        c = cos(phi_new(e,i-1));
        s = sin(phi_new(e,i-1));
        transf_matrix = [c s 0 0; -s c 0 0; 0 0 c s; 0 0 -s c];

        k_loc = (E(e)*A(e))/(elm_length_new(e,i-1))*[1 0 -1 0; 0 0 0 0; -1 0 1 0; 0
0 0 0];
        k_glob = transf_matrix' * k_loc * transf_matrix;

        m_loc = (w(e)*elm_length_new(e,i-1)/2)*[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0
1];
        m_glob = transf_matrix' * m_loc * transf_matrix;

```

```

        c_glob = 1/2 * rho_vann * Cd(e) * D(e) * (elm_length_new(e,i-1)/2)
*sqrt((c*(y_new(e,i-1)-y_new(e,i-2))/delta_t - s*(x_new(e,i-1)-x_new(e,i-
2))/delta_t)^2)*[s*s -s*c 0 0; -s*c c*c 0 0; 0 0 s*s -s*c; 0 0 -s*c c*c];

        for k = 1:4
            kglob = (e-1)*2 + k;

            for j = 1:4
                jglob = (e-1)*2 + j;

                K_total(kglob, jglob) = K_total(kglob, jglob) + k_glob(k,j);
                M_total(kglob, jglob) = M_total(kglob, jglob) + m_glob(k,j);
                C_total(kglob, jglob) = C_total(kglob, jglob) + c_glob(k,j);
            end
        end
    end
end

P_eff = (2 * M_total - K_total*(delta_t^2)) * U(:,i-1) - (M_total -
C_total*delta_t/2) * U(:,i-2) + P(:,i-1)*delta_t^2;

M_eff = M_total + C_total*delta_t/2;

% M_eff er den nye koeffisientmatrisen, så nå er det den som må modifiseres.
M_eff(1,:) = 0.0;
M_eff(:,1) = 0.0;
M_eff(1,1) = 1;
P_eff(1) = 0.0;

    for j = 1:(no_elms_bottom+1)
        index = (j-1)*2 + 2;
        M_eff(index,:) = 0.0;
        M_eff(:,index) = 0.0;
        M_eff(index,index) = 1;
        P_eff(index) = 0;
    end

% Må legge inn betingelse som gir u_y = 0 på toppen
M_total(matrix_size,:) = 0.0;
M_total(:,matrix_size) = 0.0;
M_total(matrix_size,matrix_size) = 1.0;
P_eff(matrix_size) = 0.0;

    M_eff(matrix_size,:) = 0.0;
    M_eff(:,matrix_size) = 0.0;
    M_eff(matrix_size,matrix_size) = 1.0;
    P_eff(matrix_size) = 0.0;

% I dette tilfellet blir ikke C_total fullstendig diagonal,
% så her må vi benytte det generelle uttrykket for likningsløsning.

U(:,i) = M_eff\P_eff;

%for e = 1:matrix_size;
%    U(e,i) = P_eff(e)/M_eff(e,e);
%end

% Oppdatere fysiske parametre
for e = 1:no_elms_total + 1;
    % Endring av koordinater og vinkel i hver node
    x_new(e,i) = U(2*e-1,i) + x_new(e,1);
    y_new(e,i) = U(2*e,i) + y_new(e,1);

    % Spesifisere at lina aldri kan dette under havbunnen
    if y_new(e,i) < 0
        y_new(e,i) = 0;
    end
end
end

```

```

% Holde fast lina i origo
x_new(1,i) = 0;
y_new(1,i) = 0;

for e = 1:no_elms_total;
    phi_new(e,i) = atan((y_new(e+1,i)-y_new(e,i))/(x_new(e+1,i)-x_new(e,i)));

    if phi_new(e,i) < 0
        phi_new(e,i) = 0;
    end

    elm_length_new(e,i) = sqrt((x_new(e+1,i)-x_new(e,i))^2 + (y_new(e+1,i)-
y_new(e,i))^2);

    strekk(e,i) = strekk(e,i-1) + E(e)*A(e)*(elm_length_new(e,i)-
elm_length_new(e,i-1))/elm_length_new(e,i-1);

    Fx_new(e,i) = strekk(e,i)*cos(phi_new(e,i));
    Fy_new(e,i) = strekk(e,i)*sin(phi_new(e,i));
end

```

```

end

```

Appendix E: SIMO input file

```

*****
SYSTEM DESCRIPTION SIMO
*****
'txsys, 3 lines

'LENUNI TIMUNI MASUNI GRAV RHOW RHOA
m      s      Mg      9.81  1.025 0.00125
'DEPTH DIRSLO  SLOPE
      300.0
*****
BODY DATA SPECIFICATION
*****
'CHBDY
BODY
'txbdy, 3 lines
txbdy
txbdy
txbdy
'IBDTYP
      2
'=====
BODY LOCATION DATA
'=====
'  Xglob      Yglob      Zglob      Phi      Theta      Psi
  0.00000    0.00000    0.0000    0.00000    0.00000    0.00000
'=====
BODY MASS DATA
'=====
'txmass, 2 lines
05-Oct-2007 -- 10:51:45 WV.frc WV.p2f WV.out

'xcog      ycog      zcog
  0.000    0.000    0.00
'-----
MASS COEFFICIENTS
'-----
'  rm      rixx      riyx      riyz      rizx      riyz      rizz
  800.    1.0E+05  0.000    1.0E+05  0.000    0.000    1.0E+04
'=====
HYDROSTATIC STIFFNESS DATA
'=====
'txstif, 2 lines
05-Oct-2007 -- 10:51:45 WV.frc WV.p2f WV.out

'istmod
  1
'-----
STIFFNESS REFERENCE
'-----
'  refx      refy      refz      rphi      rtheta      rpsi
  0.000    0.000    0.5      0.000    0.000    0.000
'-----
LINEAR STIFFNESS MATRIX
'-----
'kmat11      kmat12      kmat13      kmat14      kmat15      kmat16
  0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    1.0E+02    0.000    0.000    0.000
  0.000    0.000    0.000    1.000E+03  0.000    0.000
  0.000    0.000    0.000    0.000    1.000E+03  0.000
  0.000    0.000    0.000    0.000    0.000    0.000
'-----

```

```

=====
POSITIONING SYSTEM DATA
=====
1 lines
Catenary mooring
- Lines from Windmills to seafloor.
'
=====
CATENary SYSTem DATA
=====
LINE DATA
-----
'iline lichar imeth iwirun icpro
  1  1  3  0  1
'xbdy ybdy zbdy
  0  0  0
'xglb yglb xwinch
 922  0  0
'ifmopo ftime btens
  0
-----
LINE DATA
-----
'iline lichar imeth iwirun icpro
  2  1  3  0  1
'xbdy ybdy zbdy
  0  0  0
'xglb yglb xwinch
-922  0  0
'ifmopo ftime btens
  0
-----
LINE CHARACTERISTICS DATA
-----
'lichar linpty npth nptv vrange
  1  3  40  1  1
'nseg ibotco slope zglbl tmax thmin
  2  1  0 -100 8000 0
'iseg ieltyp nel ibuoy sleng fric
  1  0  20  0  600 0.8
  2  0  20  0  400 0.8
'iseg dia emod emfact uwia watfac cdn cdl
  1  0.157 7.00E+05 2.0 0.937 0.87 2.50 2.4
  2  0.187 7.00E+05 1.0 0.226 0.81 2.50 1.8
*****
SPECIFIED FORCE
*****
TXT1
TXT2
-----
'NCOMP
  1
'IMETH IDOF
  3
'XP YP ZP DV1 DV2 DV3
  0.0 0.0 0. 1. 0. 0.
'IFORTY TON TOFF P1 P2 P3
  1 200 400 100 0 0
*****
'
END
*****

```