TU Delft

Faculty of Aerospace Engineering

MEMORANDUM M-863

Cable Mechanics and Computation in B2000

P.S. Smith

January 1999

Delft University of Technology
Faculty of Aerospace Engineering
Delft, The Netherlands

Abstract

In many fields of structural engineering the behaviour of cables as structural elements is essential. One can think of suspension bridges, tethering of auxiliary crafts, the tensioning of reflectors of satelites, etc., etc.. Because cables are found in many engineering applications, the need arises for a (geometrically) nonlinear cable element that can be used in finite element analysis.

In this report a cable element will be discussed that has been implemented into the finite element code **B2000**. This element has only axial stiffness and can be used as a first approximation to the simulation of static as well as transient cable behaviour. A special difficulty with the simulation of this behaviour is that in the absence of tension, the equilibrium configuration of a cable is undefined. This peculiar characteristic is due tot the absence of bending stiffness and it poses an obstacle in the static analysis of structures where cables are used. This fundamental property, the degeneration of the stiffness of the structure as soon as cables become tensionless, requires special approaches to the solution of cable structures. Therefore, it is especially this problem that will be discussed in this report.

The developed cable element can be used for quasi-static analysis and linearized vibration analysis around stable states, but also for transient analysis. Some applications of the cable element will be presented.

Samenvatting

Het gedrag van kabels als constructie elementen kan op veel gebieden binnen de mechanica van constructies van essentieel belang zijn. Men kan denken aan bruggen, het voorttrekken van hulp voertuigen, het aanspannen van satelliet reflectoren, etc., etc.. Daar kabels in de techniek veel worden toegepast, is er een behoefte aan een (geometrisch) niet-lineair kabel element ontstaan, dat gebruikt kan worden voor eindige elementen berekeningen.

In dit afstudeerverslag wordt een kabel element besproken dat is geïmplementeerd in de eindige elementen code **B2000**. Dit element heeft alleen axiale stijfheid en kan als een eerste benadering voor de simulatie van statisch alswel van transient kabel gedrag gebruikt worden. Een speciale moeilijkheid in de simulatie van dit gedrag is dat wanneer de kabel niet onder spanning staat, de evenwichts configuratie niet gedefiniëerd is. Dit speciale kenmerk is een gevolg van het ontbreken van buigstijfheid en vormt een obstakel in de statische analyse van constructies die gebruik maken van kabels. Deze fundamentele eigenschap, de degeneratie van de stijfheid van de constructie zodra de kabels spanningsloos worden, vraagt om speciale benaderingen voor de oplossing van kabel constructies. Het is hierom dat in dit verslag met name dit probleem besproken zal worden.

Het ontwikkelde kabel element kan gebruikt worden voor quasi-statische berekeningen en voor de berekening van gelineariseerde trillingen om stabiele evenwichtsstanden, maar ook voor transiente berekeningen. Tevens zullen enige toepassingen van het kabel element besproken worden.

Acknowledgements

The development of a nonlinear cable element, which is applicable in a wide variety of problems as described in this report, would not have been possible without the help of some people who deserve my special thanks;

First of all I would like to thank Dr. ir. E. Riks for his supervision and assistance and the many useful discussions with respect to cable mechanics as well as a number of other interesting topics. Dr. ir. G. Rebel deserves my special thanks too, particularly for his explanations with respect to **B2000**. Also Ir. J. J. C. Remmers deserves my thanks for his occasional help with **B2000**. Also I would like to thank Prof. dr. J. Arbocz for his supervision and support during this period and Dr. ir. A. de Boer and ir. P. Arendsen for the introduction to **B2000** and the information with respect to the mast problem.

Furthermore I would like to thank my family for their love and patience, Bas and Joris for the good laughs, Liane, Gert and Erwan for keeping me in shape sporting, my other good friends for their care and friendship and the students and AIO's of the department and of course Annemarie for creating a pleasant atmosphere.

Contents

List of Figures

List of Symbols

A	cross sectional area
$\mathbf{C}$	damping matrix
E	Young's modulus of elasticity
$\mathbf{F}$	force
$\mathbf{K}$	(tangential) stiffness matrix
$\mathbf{M}$	mass matrix
P	total potential energy
$\mathbf{R}$	position vector deformed state; reaction force
U	strain energy
$\mathbf{U}$	global nodal displacements
V	external potential energy
W	external work
$\mathbf{a}$	local nodal displacements
$\mathbf{f}$	equilibrium equations
$\mathbf{n}$	unit outward normal
$\mathbf{u}$	displacements (continuous)
$\mathbf{r}$	position vector undeformed state
s	arc length along undeformed cable
S	arc length along deformed cable
$\mathbf{t}$	displacements (discretized) or: generalized coordinates
l	deformed cable length
l_0^*	(initial) prestressed cable length
l_0	initial (unstretched) cable length
ε	strain
ε_0	initial strain
η	path parameter
λ	load intensity factor
σ	stress
σ_0	initial stress (prestress)

Chapter 1

Introduction

Structural problems have long been analyzed by use of finite element methods. Several finite element software packages have been developed since the introduction of finite element approaches. Structural elements such as rods, beams and shells have been widely represented in these packages, particularly for linear analysis.

The cables in the structures analyzed are frequently modeled by (linear) beam elements or rod elements, where for the beam elements small values are specified for the bending stiffness. Although such approximations may work in some cases, the need exists for an element which can represent the flexible behaviour of cables. Due to this flexibility large displacements are not uncommon and the element must thus possess geometrically nonlinear properties.

Also, the unloaded cable can only resist (axial) tension loads and no bending, shear or compression loads. These characteristics make the development of a geometrically nonlinear cable element an interesting problem as alternative formulations must be found for the cable in these conditions. The geometrically nonlinear cable element with linear material properties will be developed within the finite element software package of **B2000**.

This chapter will give a short review of some cable aspects as well as a description of the data structure of the used software package **B2000**.

1.1 Introduction to cables

Cables as structural elements can be of essential importance to the stability behaviour of (cable) structures. One can think of cable stayed bridges, television towers, high voltage electricity cables and stabilization of masts. Many more applications for cables can be found in various fields of structural engineering, e.g. space technology, marine technology, civil engineering, etc.

Cables come in all variaties of cross-sectional area, number and cross sectional area of guys, material properties etc. Cables built of multiple guys in fact require an extensive separate investigation as the presence of the various guys cause an anisotropic distribution of the stiffness parameters and internal friction due to the interaction of the guys. However, such a consideration lies beyond the framework of this assignment and is left as an option for further investigation.

Instead, single guy cables with small thickness/length ($\frac{t}{l}$) ratio will be considered. For such cables a string element will be developed as a first approximation to a cable. The string will be referred to as a cable (element) throughout the report.

The cable is assumed to have only tension stiffness and no bending, shear or compression stiffness. Of course some transverse stiffness is introduced when the cable is in a state of tension. The possibility that the cable loses its (pre)tension in a specific situation presents potential numerical difficulties. The unloaded cable can take on any arbitrary form as long as the total arc-length remains unchanged. This means that the cable does not take on any definite equilibrium form.

Literature study has learnt that extensive research has been performed on cables, however mainly on linear cable elements, cables submitted to tension or cable dynamics. Few literature was found on solution approaches to the problems of singularities appearing for cables in a tensionless state.

The present code already provides an element routine for a cable with only tangential (tension) stiffness. This element routine however, is not active for specifically quasi-static analysis (**B2CONT**), and does not deal with the problem of singularity in a satisfying manner. The main objective of this report is to provide an understanding of the physical aspects of the singular behaviour of a cable due to zero or negative stress and to investigate some approaches to this problem. Chapter 6 presents some options to approach the singularity problem for quasi-static analysis.

In case of cable dynamics the singularity of the stiffness matrix is compensated by the presence of a mass matrix. Dynamic problems will be solved by transient analysis.

Finally, some suggestions are made how to approach the singularity problem by combining the non-linear quasi-static solution procedure with the transient solution procedure.

1.2 Introduction to B2000

The finite element package **B2000** has been originated by SMR Corp. from a need for a modular finite element code as a testbed. **B2000** consists of several program modules (macro-processors) which communicate independently via the central database **MEMCOM**. The modular structure of **B2000** makes it suitable for adding new independent processors. Also users can easily add new elements to the code. See also figure 1.1.

The **Input Processor** reads the model data and generates a **B2000** data base by creating the appropriate **MEMCOM** data sets. These data sets are required for running several macro processors some of which are mentioned below:

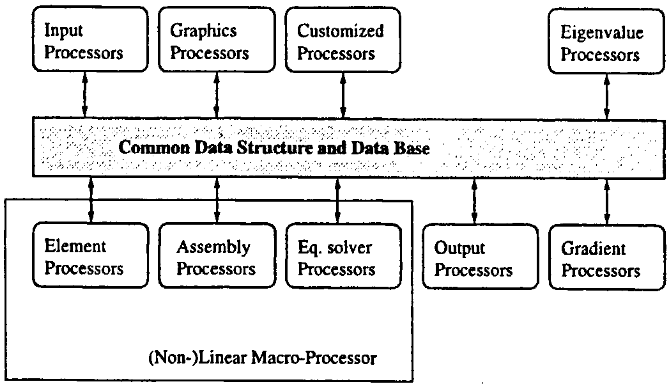


Figure 1.1: The basic structure of B2000

B2LIN linear solver

B2CONT non-linear (quasi-static) solver

B2TRANS transient solver; recently modified and implemented

The programming language of the code is *FORTRAN 77* and *ANSI C*. However as the code has a modular design **B2000** is not limited to these languages and can easily be extended by modules written in other programming languages.

Finally, the results obtained from a numerical analysis can be post-processed with the post processor **B2BASPL**. **B2BASPL** enables graphical visualization of the results by colour plots and diagram plots. Additionally, it is possible to transform a **NASTRAN** bulk data deck to a **B2000** input deck and vice-versa using the processor **B2NAS**.

The finite element code **B2000** is currently used as a testbed by the *Delft University of Technology*, the *Twente University of Technology*,the *Swiss Federal Institute of Technology (EPFL)*, the *National Aerospace Laboratory, NLR*, the *Deutches Zentrum für Luft- und Raumfahrt* and the *Centro Italiano Ricerche Aerospaziale.*

Chapter 2

Cable mechanics

Nonlinear analysis generally involves geometric (strain-displacement) nonlinearity and/or material (stress-strain) nonlinearity, [17]. Taking into account large displacements and assuming small strains, a cable element with nonlinear geometric properties and linear material properties, satisfies the requirements for most nonlinear analyses of cable-structures. Section 2.1 will illustrate the geometric nonlinear aspects of a cable by showing the nonlinear load-displacement relationship for a two-link cable construction.

Small-step incremental solution methods like the *path following technique* can only be applied to solve the nonlinear equilibrium equations for the unknown displacements at points where the tangential stiffness matrix is not singular, but the stiffness matrix becomes singular at the instant the cable becomes tensionless. This means that procedures are necessary to remove the singularity. Please note that the singularity pertains to all nodal values that belong to the cable with no (pre) tension. The multiplicity of the singularity in the total stiffness matrix is then usually larger than one.

Section 2.2 presents a qualitative approach to the physical aspects of a singular cable. The main objective of this approach is to acquire a fundamental understanding of the physical cable behaviour in relation to the stability behaviour of beams. Section 2.3 presents a brief discussion of singularities occurring in a discretized cable.

Finally, section 2.4 presents a description of the cable properties and behaviour in specific situations, illustrated by some simple examples. Section 2.4 sets the boundary conditions for the element that has to be developed.

2.1 Geometric nonlinearity

For a wide range of structural problems, both displacements and strains are small. This means that during the loading and deformation process the geometry of the structure remains basically the same, allowing the problem to be solved by linear solution procedures.

However, cable-structures submitted to static (or dynamic) loading, may often result in large displacements without actually causing large strains. This allows

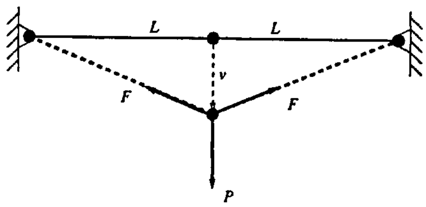


Figure 2.1: Two-link cable construction submitted to transverse loading to illustrate the geometric stiffness effect

the discussion (in this report) to be specifically restricted to geometric nonlinearity, leaving the material properties to be approximated by linear strain-stress relations.

As most construction problems involve conservative loads, the discussion in this report is also restricted to problems with *conservative* load systems. This means that only problems are considered where forces themselves are derivable from a potential function. As a consequence, the final deformed configuration or state of a structure submitted to such a load system is not dependent on which *load path* has been followed. An example of this aspect is discussed in more detail in section 2.4, part *(v)*. For the analysis of structures submitted to loads dependent on the deformations, proper consideration must be given to the displacement-dependent load term.

This section discusses the geometric nonlinear aspects of cable behaviour illustrated by a simple two-link structure as shown in figure 2.1. Obviously, in this initial (straight) configuration the cable construction is incapable of resisting the imposed transverse load P; no reaction forces can occur in the cables such that (vertical) equilibrium is satisfied.

As a consequence we are initially dealing with a *mechanism* in the v-direction The dashed lines in figure 2.1 represent a deformed configuration. As the mid-node undergoes vertical displacement (v), the reaction forces in the cable elements now acquire vertical components which can equilibrate the imposed external load P. Hence, the resistance to transverse loading is introduced by a change of geometry and originates from axial straining.

To illustrate the nonlinear relationship between the external load P and the transverse displacement v of the mid-node, consider figure 2.2. For this purpose the nonlinear relationship between Δu and v will be derived first by considering the projection of the deformed configuration (dS) onto the undeformed configuration (ds).

This is achieved by considering a 'tensionless' case, where node 3 is only locked in the transverse direction and a displacement of the mid-node will result in a horizontal displacement Δu of node 3 such that the cable elements do not undergo any elongation. The lengths of the cable elements in the deformed state thus remain L. The projection onto the undeformed state is expressed by:

$$2Lcos\phi = 2L - \Delta u \qquad (2.1)$$

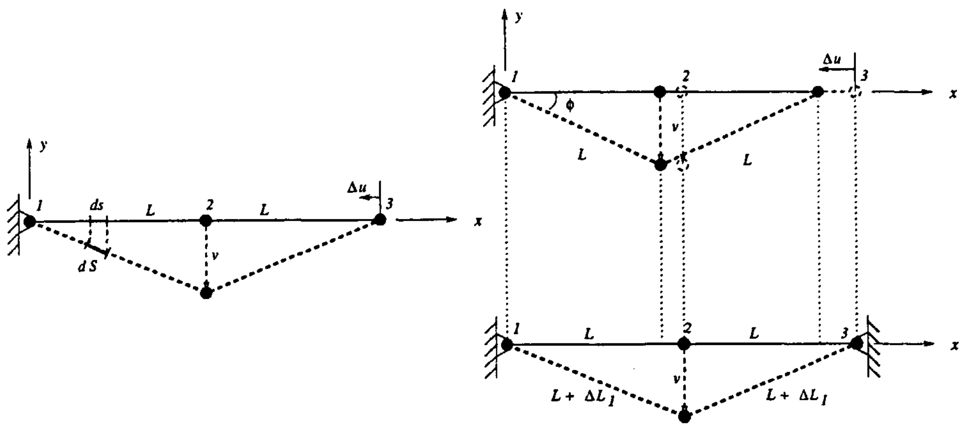


Figure 2.2: Illustration of geometric nonlinear deformation

or, rewritten:

$$\Delta u = 2L(1 - cos\phi) \tag{2.2}$$

The 'elongation' of a cable element with respect to the projection onto the undeformed state is:

$$\Delta L = L - Lcos\phi \quad <=> \tag{2.3}$$

$$\Delta L = L(1 - cos\phi) = \frac{1}{2}\Delta u \tag{2.4}$$

This corresponds to the elongation of a cable element if node 3 were fixed at $x = 2L - \Delta u$. Hence, for the undeformed length one can write:

$$L_0 = L_d - \Delta L = L - \frac{1}{2}\Delta u \tag{2.5}$$

and the deformed length:

$$L_d = L_0 + \Delta L = L \tag{2.6}$$

Next, consider the real case in the lower right part of figure 2.2. All translational degrees of freedom of node 3 are fixed and the undeformed cable length is now L. Displacement of the mid-node over a distance v results in stretched cables with deformed lengths $L + \Delta L_1$. It is easily recognized that consistent substitution of these expressions into eqs.(2.5) and (2.6) yield the same relationship for the elongation ΔL_1 and the longitudinal displacement Δu, see eq. (2.4).

Eq.(2.2) can be expressed in terms of the vertical displacement v by realizing that

$$\sin\phi = \frac{v}{L} \tag{2.7}$$

This yields the following expression:

$$\Delta u = 2L\left(1 - \sqrt{1 - \left(\frac{v}{L}\right)^2}\right) \tag{2.8}$$

As can be seen from eq.(2.4) the displacement Δu is proportional to the elongation ΔL which in turn is proportional to the strain;

$$\varepsilon = \frac{\Delta L}{L_0} \tag{2.9}$$

For linear elastic material and constant cross-sectional area, the strain is proportional to the cable stress by Hooke's law:

$$\sigma = E\varepsilon \tag{2.10}$$

Having proven the nonlinear relationship between Δu and v and stated that Δu is proportional to the reaction forces F, it has thus been proven that P is a nonlinear function of v;

$$P = f(v^2)$$

Obviously, linear solution techniques cannot be used to solve the (nonlinear) equilibrium equations, corresponding to the geometric nonlinearity. Contrary to linear problems (unique solution situation), the solution obtained for a nonlinear problem may not be the solution sought. As more solutions exist, one may find a physically insignificant solution. The nonlinearity usually requires a small-step incremental approach in order to follow the solution path and in this way gain physically significant and accurate results. Without *path following techniques* it is in general not possible to obtain the solution in a controlled way.

The finite element code of **B2000** provides a quasi-static (nonlinear) solver (**B2CONT**), which solves static nonlinear problems by a *path following technique* , [14], [15]. A short description of this *path following technique* is given in section 3.2, followed by a description of the nonlinear solver **B2CONT** in section 3.3.

2.2 An interpretation of (singular) cable behaviour

The behaviour of a cable under tension and in particular, in circumstances where tension is lost, can be explained in the context of classical stability theory [2].

Consider a slender beam of the same dimension as the cable tethered between two supports of which one can only move in the axial direction. (See figure 2.3). The beam has the same axial stiffness in tension and compression. Compression can also be achieved in a cable if we prevent the cable from having any lateral displacements. The only difference between the cable and beam is that the beam has bending stiffness and the cable has not.

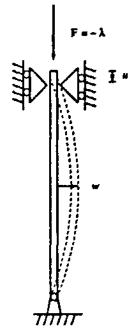


Figure 2.3: Column buckling example

beam under axial compression
The first most general situation to be discussed in the context of stability of an element submitted to axial compression will be an imperfect beam-column under axial compression, see figure 2.3. As the beam possesses both (axial) compression and bending stiffness, the beam will compress a small axial distance u and bend a given distance w (lateral displacement amplitude) determined by the bending stiffness (EI) of the beam. See the dash-dot lines in figure 2.4.

In case of an ideal beam however, the beam will follow the *primary equilibrium path* ($w_I(\lambda) = 0$) with only axial displacements, which becomes unstable after the first bifurcation point. At this first bifurcation point a stable *secondary equilibrium path* intersects the primary branch. The solid lines in figure 2.4 represent the stable equilibrium paths whereas the dashed lines indicate the unstable paths.

The corresponding load-displacement curves are presented in figure 2.4, where λ represents the load parameter. The figure shows the first few secondary branches (buckling) initiated at the bifurcation points. Obviously, specifically the first bifurcation point is of interest for stability considerations as it initiates the post-buckling behaviour.

Note that for a beam with high bending stiffness a larger compressive force is required to initiate buckling, than for a beam with low bending stiffness.

cable under axial compression
The singular behaviour of a cable submitted to axial compression can be explained by replacing the column in figure 2.3 by a cable, using the concept of a gradually diminishing bending stiffness of the beam so that in the limit of EI approaching zero, the beam becomes a cable. When the bending stiffness

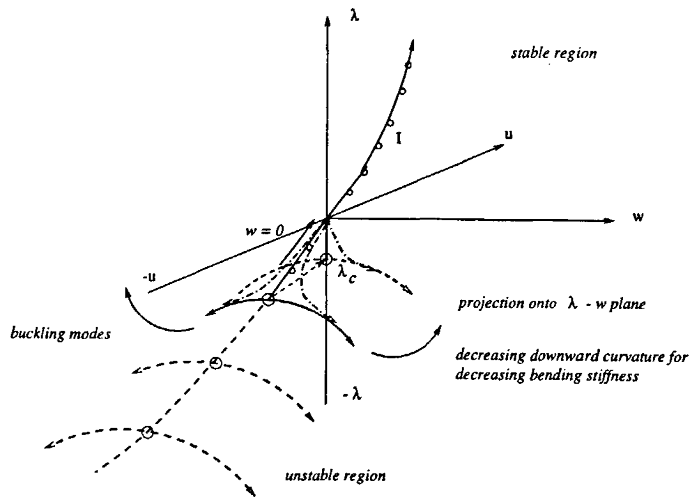


Figure 2.4: Bifurcation into buckling modes of a beam under axial compression

of a beam in compression (figure 2.3) approaches zero, also the buckling loads will approach zero. In other words the bifurcation point at the value of the critical load λ_c will come closer and closer to the undeformed and unloaded state $(w = u = 0, \lambda = 0)$.

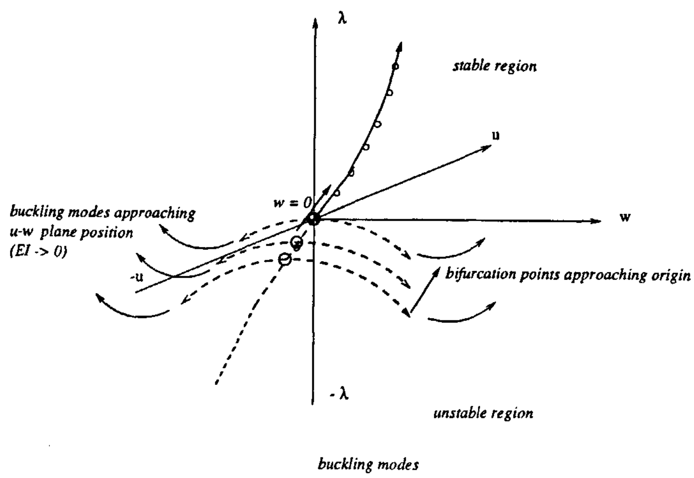


Figure 2.5: Bifurcation points approaching origin for a beam under axial compression with gradually diminishing bending stiffness

This is not only true for the first bifurcation point along the pre-buckling state ($w_I = 0$) at λ_c, it is also true for all the (infinite) UNstable bifurcation points that are located beyond this point. This means that in the limit of zero bending stiffness, the undeformed state $w = u = 0$ at $\lambda = 0$, becomes an accumulation of stable and unstable bifurcation points of an (in principle) infinite multiplicity. (See figure 2.5).

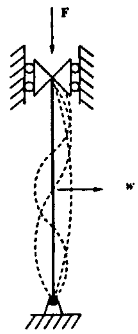


Figure 2.6: Infinite buckling modes for a cable under axial compression with unlocked transverse degrees of freedom

Because a cable cannot sustain compression, (unless it is completely restrained in the transverse direction) the unstable bifurcations at λ_c=0 will take place in the plane $\lambda_c = 0$ and the post-buckling paths in this plane are undetermined. 'Buckling' can thus take place at λ_c in any form that is compatible with the cable, i.e. as long as the kinematic condition of unchanged total cable-length is satisfied. (See figures 2.6 and 2.7).

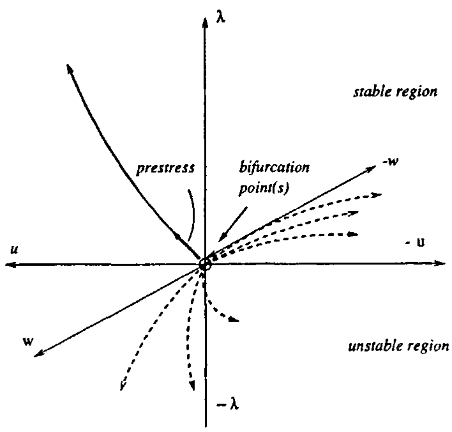


Figure 2.7: Bifurcation into the buckling modes of a cable under axial compression

prestress effect
Clearly, even when a positive load increment is imposed (tension), the computation will meet difficulties at the start of the analysis. One way to avoid this starting problem is to start from a given pre-stressed situation. Giving a prestress means that the cable is brought into a stable state that is part of a stable branch connected with the undeformed state. In this stable region displacements in the transverse direction will not lead to instability, while from the figure it becomes obvious that in the unstable region transverse displacements will lead the cable into one of the many (unstable) paths ($w \neq 0$). The effect of prestress is illustrated in figure 2.7.

However, as will be discussed in subsequent chapters, pre-stressing a cable is not always a practical option in order to obtain the desired results and in some cases it leads to modeling problems.

2.3 Transverse singularity

In the previous section attention was given to the physical stability aspects of a continuous cable under compression. This section will extend on the singularity aspects of the discretized cable.

Recall from the previous section the (continuous) cable submitted to compression. Each point along this cable has transverse degrees of freedom. As a continuous cable is considered here, we are dealing with an infinite number of (transverse) degrees of freedom. Consequently, if the cable is not *completely* restrained in the transverse directions the cable will become unstable due to its negligible bending stiffness. (See section 2.2). By restraining all transverse displacements along the cable, i.e. , by placing the cable in e.g. a 'tube' the system is entirely stabilized (!), meaning that the cable is then capable of resisting the compressive load.

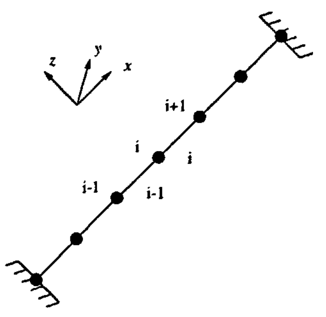


Figure 2.8: Discretized cable section

Next, consider the discretized cable, see figure 2.8. All nodes in between the two outer nodes, which are connected to the 'fixed' world, will be referred to as 'interior nodes' in the sequel. Discretization of the cable into a finite number of elements results in a finite number of nodes along the cable. The behaviour of the cable is then described by determining the displacements of each node by some numerical solution procedure. (See also chapter 3).

This means that the infinite number of degrees of freedom of the continuous cable has now been reduced to a finite number of degrees of freedom at the nodes. Consequently, in order to stabilize the discretized cable when submitted to compression, it is required to restrain the (finite number of) transverse degrees of freedom of the interior nodes.

However, in many cable-structures these degrees of freedom are not or may not be restrained resulting in a multiplicity of singular values equal to the number of (un-restrained) degrees of freedom that represent the transverse deformation of the cable. These singular values appear in the stiffness matrix at corresponding positions yielding a singular stiffness matrix.

The tensionless state of a cable, including the involved singularities as described above, may occur in different situations. The situations most likely to be encountered, will be described in section 2.4.

2.4 Cable properties

To set out the borders for the class of structural problems that have to be solved by use of the cable element, an evaluation must first be made of the problems that may be encountered. The following conditions must be satisfied:

(i) The load-displacement curve is nonlinear (geometric nonlinearity). This requires proper evaluation of the required (nonlinear) equilibrium equations and tangential stiffness matrix. See also chapter 3.

(ii) The unstretched lengths of the cable members are initially unknown. These must be obtained from the initial stresses if these are given.

(iii) The option must be available of applying the external loads by use of prescribed force or prescribed displacement. Obviously, the results must be consistent for either option.

(iv) Some cable members may become slack during the deformation process as it assumes it's deformed shape. The tensionless state of a cable is an undefined configuration and hence a formulation must be found to enable continuation of the computation.

(v) If there is no tension in the cable members in the initial state , the initial state may be undefined and can not be used as initial state in the computation.

(vi) As many constructions consist of different components, another requirement is that the cable element must be compatible with other structural elements such as beams and shells.

To enable proper analysis for the situations described above, methods must be found to overcome these problems. In order to do so a more extensive evaluation of these problems is discussed in this section.

The implications involved with the actual implementation into the element routine are discussed in detail in chapters 5 and 6.

(i) *provide* **B2CONT** *routine*
Obviously, the most elementary condition the element (routine) must obey is that it must provide the **B2CONT** routine with the expressions required to perform the continuation analysis, i.e. the tangential stiffness matrix and the nonlinear equilibrium equations (see chapter 3). Most importantly, the cable element must be implemented such that singular situations are dealt with in a satisfying manner. The main aspects that have to be taken into consideration for this purpose are described below.

Also, in order to enable dynamic analysis such as linearized vibrations around stable states and transient analysis a mass matrix definition must be provided.

(ii) *pre-stress option*
The importance of pre-stress as a very acceptable and obvious way to avoid singularity problems at the start of an analysis has become clear in the section 2.2. Therefore it is essential that a pre-stress option is available. By Hooke's law for linear elastic materials the following stress-strain relationship holds in the pre-stressed state:

$$\sigma_0 = \varepsilon_0 E \tag{2.11}$$

where for the strain:

$$\varepsilon_0 = \frac{l_0^* - l_0}{l_0} \tag{2.12}$$

see also the strain definition in section 4.2. In the initial pre-stressed state, l_0^* represents the deformed length due to the pre-stress and $l_0^* - l_0$ the corresponding elongation. When performing an analysis with an additional external load the analysis will start from this initial pre-stressed state and the cable will deform to a total deformed length l, resulting in a total stress of

$$\sigma = E\left(\frac{l - l_0}{l_0}\right) \tag{2.13}$$

Comparing this to eq. (2.11) the additional term due to the applied external load is seen to be:

$$\sigma - \sigma_0 = E\left(\frac{l - l_0^*}{l_0}\right) \tag{2.14}$$

Pre-stress can be applied to an element by defining an initial cable length l_0 or directly by defining an initial (pre)stress σ_0. To allow proper evaluation of the deformed configuration of a cable construction submitted to external loads, it is desirable to have the total stresses and displacements computed explicitly. This means that independently of whether the pre-stressed state is prescribed by l_0 or directly by σ_0, l_0 and l have to be available to enable computation of the total stress, see eq. (2.13).

The most convenient way to achieve this is to compute the initial (unstressed) length l_0 for both cases. Hence, when the pre-stressed state is prescribed by l_0, this value is stored directly for further computation. However, when σ_0 is given, l_0 is first determined from

$$l_0 = \frac{l_0^*}{\frac{\sigma}{E} + 1} \tag{2.15}$$

where-after it can be stored for further computation. The total deformed length l can be computed from the given node co-ordinates and the computed displacements, see section 4.5.

The equilibrium equations and the tangential stiffness can thus be described unambiguously in the remainder of the analysis in terms of the initial length l_0.

Obviously, the results of identical models submitted to identical load-cases must coincide whether initial stresses or their corresponding initial lengths have been specified. A similar condition holds for the option of applying load-cases by prescribed force or prescribed displacement, as will be discussed in the next part.

(iii) *prescribed loads versus prescribed displacements*
Another essential option that must be available through the implementation of the cable element is the definition of load-cases by prescribed forces and prescribed displacements. It needs no argumentation to realize that similar to the pre-stress option described above, the results must coincide for identical problems, independent of which choice has been made to define the load-case. When the external force has been prescribed, **B2CONT** will determine the corresponding nodal displacements and vice-versa. The relationship between prescribed loads and prescribed displacements will be treated in section 3.2.

(iv) *cables as part of a construction*
By quasi-static analysis of a multi-cable construction like the mast of a sailing ship or a cable suspended bridge submitted to external forces, one can determine the deformations and internal stresses. However, as these constructions often consist of several cables discretized into several cable elements, it is not unlikely that during the deformation process one or more cable elements become tensionless (for a number of load-cycles) and hence no longer contribute to the stiffness of the structure as a whole.

To illustrate this, consider a simple cable construction as presented in figure 2.9. The cables are all given a small prestress to initialize the computation. During the deformation process the cables 1 and 2 are stretched into a state of tension, while cable 3 becomes tensionless and no longer contributes to the stiffness of the construction. This state is represented by position 'II' in the figure.

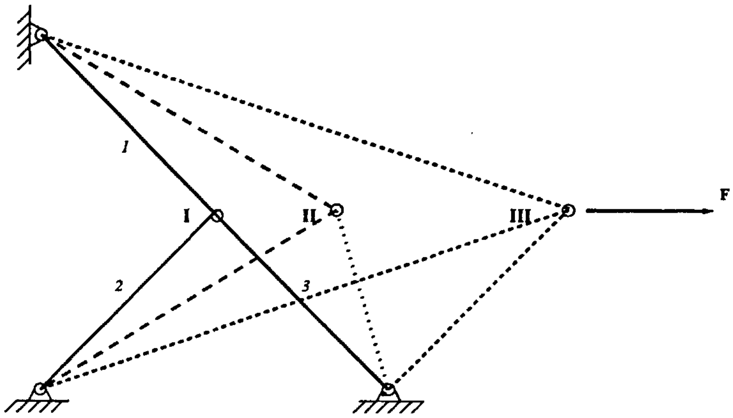


Figure 2.9: Cable construction illustrating negative stress in one cable during the deformation process

Hence:

$$l_1 > l_{0_1}$$
$$l_2 > l_{0_2}$$
$$l_3 < l_{0_3},$$

where l_i represent the deformed lengths, and l_{0_i} the initial undeformed (and tensionless !) lengths.

As long as the remaining parts of the structure can carry the loads, the construction will not collapse. More precisely stated, the 'slack' cables merely carry their own weight due to the presence of gravity, adapting a sagged configuration. As the weight of the cables is generally small compared to the external loads involved, the effect of gravity and the corresponding small tension in the slack cables can be neglected. This is an acceptable simplification which reduces the required computation time and the time required to define the model. Numerically however, this leads to the singularity problems as described previously.

Obviously, it is not desirable for the computation to stop due to one or a few tensionless cables, while the construction as a whole is still capable of resisting the external loads. If the incremental increase of the external load **F** is continued, the structure will continue to deform accordingly (position 'III'). As a consequent, cable 3 will re-adapt a state of tension ($l_3 > l_{0_3}$) for sufficiently large deformations, see figure 2.9. Naturally, such large deformations will only occur for cables with very low tension stiffness. The example is merely intended to illustrate a possible deformation process of a multi-cable construction.

To establish continuation of the computation for similar cases, several options were considered. An evaluation of these options will be discussed in the sequel of this report.

(v) *problem of initial unloaded condition*
Consider a simple example of a cable suspended between two single supports submitted to gravity, resulting in cable sag (see figure 2.10). This is the most elementary problem involving singularity that has to be solvable. The initial length and mass distribution of the cable are given. The distance between the supports is fixed and equal to the initial length l_0 of the cable. This sagged position can be determined by incrementally increasing the gravity load from *zero* to g ($= 9.81$ $[m/s^2]$).

However, as was stated in section 2.3 this will result in singularity problems at the start of the computation as the cable is initially stress-free. Therefore it is necessary to define a prestress prior to imposing the external loads.

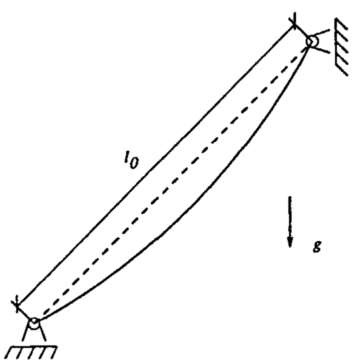


Figure 2.10: Cable submitted to gravity resulting in cable sag

With the available tools provided by the **B2000** code and the present element routine, this problem can be solved by defining an indirect load path which results in the same end configuration. This is possible since only conservative load-systems are considered.

See figure 2.11. To enable the computation to start, the cable is prestressed by a small prescribed displacement $\Delta\mathbf{u}$. (Note that a prestress can also be established by prescribed loads. In this case it is more convenient to use prescribed displacements). Transverse stiffness is now introduced due to the pre-stress and thus the gravity load can be applied incrementally. By finally reducing the initially introduced elongation $\Delta\mathbf{u}$ back to zero, the desired end configuration as defined in figure 2.10 is obtained.

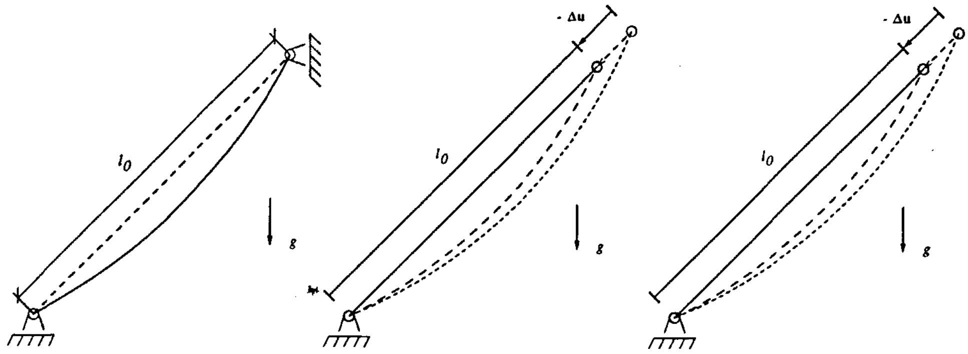


Figure 2.11: Computation of cable sag using prescribed displacements for initial elongation

Obviously, the described method is rather inconvenient and difficulties arise as the model is defined. Also, in many cases the cables are part of a larger structure and thus connected to other parts of the construction.

Executing an analysis with an initially pre-stressed state will thus result in some practical geometry definition problems. For these reasons it is desirable to enable direct superposition of the external loads to an initially unstressed cable with given geometry. Several options to achieve this will be discussed in the chapter 6.

(vi) *compatibility with other elements*
Finite element models of constructions generally consist of various different elements like rods beams and shells. Hence, it is essential that the element to be implemented is compatible with the other elements present in the **B2000** code.

Note that contrary to the situations described in (*v*) and (*iv*), some properties described above do not necessarily involve negative stress situations ((*i*), (*ii*), (*iii*), (*vi*)). The modifications required to establish proper cable behaviour for tension cases will first be discussed in chapter 5. The tensionless states require a more extensive evaluation. Chapter 6 will present several options to overcome the problems involved, illustrated by example problems where necessary.

Chapter 3

Solution method

Typical phenomena occurring with the analysis of stability behaviour of structures are *critical points* and *bifurcation points*. The locations of these points in the solution sets have been of considerable interest for many years and so several solution techniques have been developed to locate these special points on the solution curve.

One solution technique that is particularly interesting for the non-linear analysis of finite element methods is the *path following technique*, [14], [15]. This technique is based on continuation methods, which make use of predictor-correction procedures. As the technique follows the solution curve, the limit points and bifurcation points can be easily recognized.

In section 2.1 an illustration of the non-linear relationship between the external force and the corresponding displacement was given. It was stated that solving the non-linear equilibrium equations would require a small step incremental solution technique to follow the path dependency. The nonlinear solver **B2CONT** provided by the **B2000** source code is based on the *path following technique.* The *path following technique* is described in section 3.2 and the **B2CONT** macro processor in section 3.3. A short description of the finite element method will first be given in section 3.1.

3.1 The finite element method

Consider an arbitrary continuous elastic body submitted to some external conservative loading system, see figure 3.1. The external load $\mathbf{F}$ is a continuous function of the position vector $\mathbf{x}$. As a result of the imposed external loading, the continuum will undergo some deformation $\mathbf{u}$, which will also be a continuous function of the position vector $\mathbf{x}$.
Hence, the problem can be represented by the equilibrium equations and the boundary conditions;

$$\mathbf{f}(\mathbf{x};\lambda) = \mathbf{0} \quad + \textit{boundary conditions} \tag{3.1}$$

The load can be imposed either by prescribed force or displacement by incrementally increasing λ:

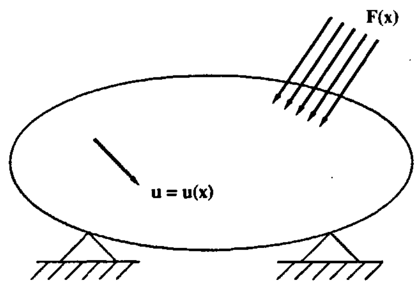


Figure 3.1: Continuum submitted to conservative loading system

$$\mathbf{F} = \lambda \mathbf{F}_0 \tag{3.2}$$

$$\mathbf{u} = \lambda \mathbf{u}_0 \tag{3.3}$$

Note that if $\mathbf{F}_0$ and $\mathbf{u}_0$ indicate the desired end values, λ must be increased from 0.0 to 1.0. Both options must be available in the code. Prescribed displacements are treated in subsection 3.2.5.

The possibilities of solving continuous problems in an exact way is usually limited by the available mathematical techniques. With the introduction of the digital computer, new possibilities to solve *discrete* problems were born even for large numbers of degrees of freedom. For this purpose the continuous body is discretized into a number of elements such that the discretized structure approximates the continuous structure within acceptable margins. This requires also discretization of the imposed external loads ($\mathbf{F}_i$) and the displacements ($\mathbf{t}$), see figure 3.2

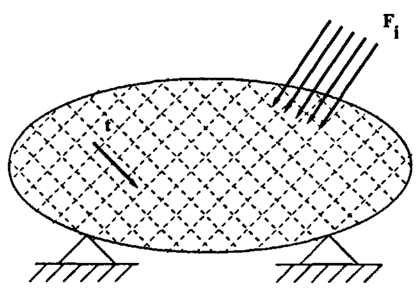


Figure 3.2: Discretized body submitted to a conservative loading system

The method of finite elements is based on the principle of virtual work. Pointwise equilibrium is replaced by a weak form. Equilibrium is then satisfied on the average over a certain small domain of the body. Such a (small) domain can now be seen as an element. It is sufficient to develop only one single element capturing all the required properties.

A discrete model of the body results from an assemblage of these elements. This model has a finite number of degrees of freedom which are used to describe the behaviour of the body.

The equilibrium equations of the elements are then assembled into one matrix equation representing the set of equilibrium equations for the complete structure [12]. The problem in its discretized state is thus defined by:

$$\mathbf{f}(\mathbf{t};\lambda) = \mathbf{0}, \quad \mathbf{f} \in \mathbb{R}_N \ , \mathbf{t} \in \mathbb{R}_N \quad + \textit{boundary conditions} \tag{3.4}$$

These equilibrium equations form the basis for the solution technique discussed in the following section.

3.2 Path following technique

3.2.1 Equilibrium equations

The deformation of the body is described by a set of N deformation parameters. The equations governing this deformation are given by the (discretized) equilibrium equations (recall section 3.1);

$$\mathbf{f}(\mathbf{t};\lambda) = \mathbf{0}; \tag{3.5}$$

where

$\mathbf{f} = [f_1...f_N]^t$ = first variation

$\mathbf{t} = [t_1...t_N]^t$ = computational degrees of freedom

λ = generalized load factor

N = total number of unknowns $\mathbf{t}$

The corresponding load-displacement curves can be obtained as curves in a (N+1) dimensional space spanned by the (N) deformation parameters and the load intensity factor λ. Assuming the undeformed configuration corresponds to the condition

$$(\mathbf{t};\lambda) = (\mathbf{0};0), \tag{3.6}$$

the solution curve is described by eq. (3.5), see figure 3.3

3.2.2 Parameterization of solution curves

The equilibrium equations (3.5) represent N relations for N+1 unknowns, i.e. $[t_1...t_N]$ and λ. Obviously, one extra relation is required in order to obtain a solvable set of equations from which a specific point on the solution curve can be determined. This *auxiliary surface* is represented by $f_{N+1}(\mathbf{t};\lambda)$=0. A specific

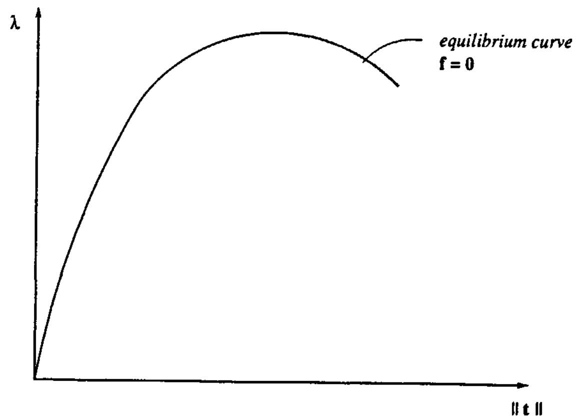


Figure 3.3: Equilibrium curve

point on the solution curve can now be obtained by solving the following set of equations:

$$\begin{aligned} \mathbf{f}(\mathbf{t};\lambda) &= \mathbf{0} \\ f_{N+1}(\mathbf{t};\lambda) &= 0 \end{aligned} \tag{3.7}$$

Several choices are possible for the auxiliary surface f_{N+1}:

(i) The simplest choice is that of a horizontal auxiliary surface, i.e. parallel to the deformation axis:

$$f_{N+1} = \lambda - \eta = 0 \tag{3.8}$$

where η is the prescribed value of the load. This is not always a convenient choice, as can be seen from figure 3.4.

If the solution curve has a limit point L, the auxiliary surface does not intersect the solution curves $\mathbf{f}(\mathbf{t};\lambda) = \mathbf{0}$ beyond the limit point. (i.e. if $\lambda = \eta$ represents a load level higher than the limit load level λ_L). Consequently, the part of the solution curve beyond the limit point cannot be computed with this definition of the auxiliary surface.

(ii) A better choice would be to adapt the direction of the auxiliary surface to the direction of the solution curve, see figure 3.5. The distance $\Delta\eta$ is an approximation for the path length ΔS between the points A and B. The unit tangents at the points A and B on the solution curves are represented by $\mathbf{n}_A^*$ and $\mathbf{n}_B^*$ respectively.

For sufficiently small step-size it is now possible to follow the equilibrium curve beyond the limit point. For this purpose the auxiliary surface is defined as follows:

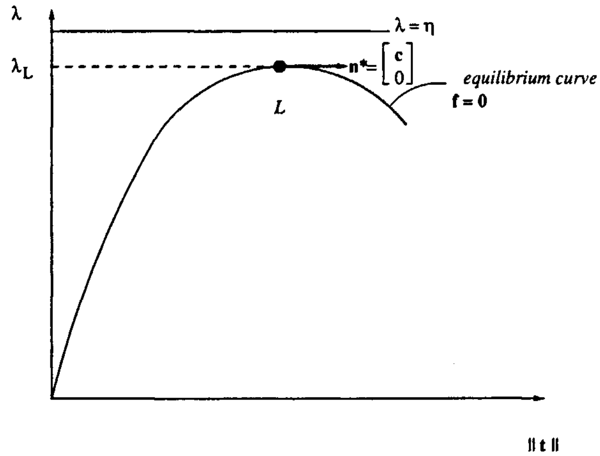


Figure 3.4: Limit point problem for auxiliary surface choice (i)

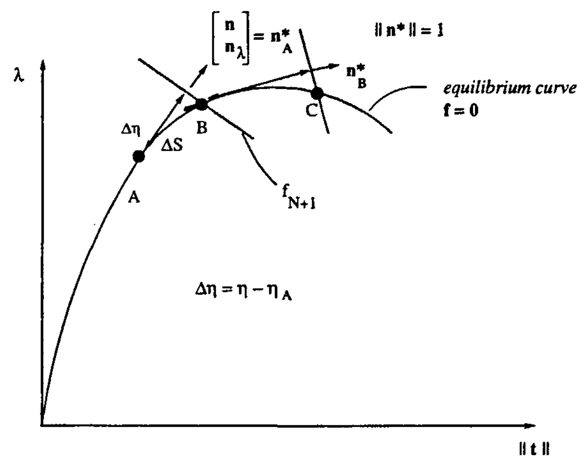


Figure 3.5: Limit point problem for auxiliary surface choice (ii)

$$f_{N+1} = \mathbf{n}^T(\mathbf{t} - \mathbf{t}_A) + \eta_\lambda(\lambda - \lambda_A) - (\eta - \eta_A) = 0 \qquad (3.9)$$

The basic idea behind this continuation method, based on prediction-correction principles is described in subsection 3.2.3.

3.2.3 Continuation procedure

Continuation methods generate a set of solutions representing points on the solution curve of interest. Subsequently, the obtained information is used to construct an approximation curve through these points. The main part of this procedure is the generation of this sequence of solution points itself. The basic procedure used by this method is illustrated in figure 3.6.

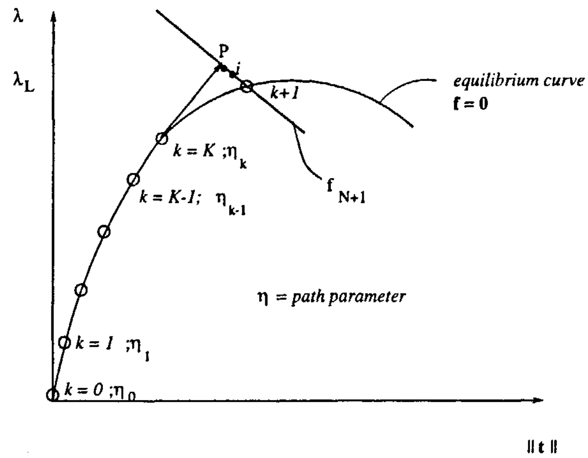


Figure 3.6: Continuation procedure

Some solution points k = 0,...,K are given on this solution curve. These points are computed in a stepwise prediction-correction manner as will be described in the sequel. For this purpose the following vector of unknowns is introduced for convenience:

$$\mathbf{x} = \begin{bmatrix} \mathbf{t} \\ \lambda \end{bmatrix} \tag{3.10}$$

prediction
The first step is a prediction of a point P at a given solution point *k*. There are various methods to make such a prediction. The most important methods will be discussed briefly below.

* *Euler prediction*:
This prediction method requires the tangent to the path, which is obtained by differentiation of eqs. (3.7) with respect to the path parameter η, yielding:

$$\begin{aligned} \mathbf{f}_{,\mathbf{t}}\dot{\mathbf{t}} + \mathbf{f}_{,\lambda}\dot{\lambda} = \mathbf{0} \\ f_{N+1,\mathbf{t}}\dot{\mathbf{t}} + f_{N+1,\lambda}\dot{\lambda} = -\dot{f_{N+1}} \end{aligned} \tag{3.11}$$

with

$$()_{,\mathbf{t}} = [()_{,t_1} \cdots\cdots ()_{,t_N}]$$
$$\dot{()} = \frac{\partial}{\partial \eta}$$

The prediction P is then determined from the intersection of the auxiliary surface with the chosen step of the path parameter in the direction of the tangent.

* *Extrapolation methods*:
The predictions constructed by extrapolation methods are based on previously obtained solutions. A prediction can be made by either linear extrapolation based on the two previously obtained solution points or by quadratic extrapolation based on the last three obtained solution points. Note that the method of quadratic extrapolation tries to include the curvature of the solution curve into the prediction.

correction
From figure 3.6 one can see that the prediction is generally not located on the curve. Hence, the equations (3.7) are not satisfied by a small residual. This situation can be represented by:

$$\begin{aligned} \mathbf{f}(\mathbf{t},\lambda) &= \mathbf{r} \\ f_{N+1}(\mathbf{t},\lambda;\eta^{k+1}) &= r_{N+1} \end{aligned} \tag{3.12}$$

By use of an iteration procedure the prediction will be corrected to the solution curve. The successive corrections are forced to stay on the auxiliary surface. When the point $i = I$ is converged to a point in the close vicinity of the solution curve within a given error margin set by the convergence criteria, the obtained point is accepted as the next solution point (k+1) and used for the next prediction step, etc. The iteration method used for this purpose is the *Newton method.* Basically, this method iterates until the residuals $\mathbf{r}$ and r_{N+1} in eqs. (3.12) approach zero and thereby indicates that eqs. (3.7) are solved. A detailed description of this method can be found in ref.[15].

Thus in general, the *path following technique* requires the following expressions:

- the equilibrium equations,

$$\mathbf{f}(\mathbf{t};\lambda) = \mathbf{0} \tag{3.13}$$

- an auxiliary surface,

$$f_{N+1}(\mathbf{t};\lambda) = 0 \tag{3.14}$$

- The prediction step requires the tangent to the path. This expression is obtained by differentiation of (3.5) with respect to the path parameter η. This introduces the term $\mathbf{f}_{,\mathbf{t}}$ which represents the *tangential stiffness matrix.* Hence, the following expression is also required:

$$\mathbf{f}_{,\mathbf{t}} \quad ()_{,\mathbf{t}} = \frac{\partial}{\partial \mathbf{t}} \tag{3.15}$$

3.2.4 Special points

There are two distinctive types of equilibrium states along the solution curve that have a special meaning; limit points and bifurcation points. These points correspond to the instability phenomena as known from the classical elastic stability theory. Both points will be discussed briefly in this subsection. Only simple points will be considered, i.e. points at which only a single singularity exists.

limit points
A limit point occurs whenever the load factor λ reaches a maximum value. See also figure 3.4. As can be seen from the figure the unit tangent $\mathbf{n}^*$ in the limit point L has no component in the direction of the load parameter λ. From eqs. (3.11) this results in:

$$\mathbf{f}_{,t}\mathbf{c} + \mathbf{f}_{,\lambda} \cdot 0 = \mathbf{0} \qquad \mathbf{f}_{,t} = \mathbf{K}_T \tag{3.16}$$

As a consequence the stiffness matrix at the limit point becomes singular. But, by choosing an appropriate auxiliary surface (3.14) this problem can be avoided and the system of equations can be solved.

bifurcation points
A simple bifurcation point occurs when two solution curves intersect. This means that two unit tangents with different directions ($\mathbf{n}^*{}_1$ and $\mathbf{n}^*{}_2$) exist at such a point. See also figure 3.7.

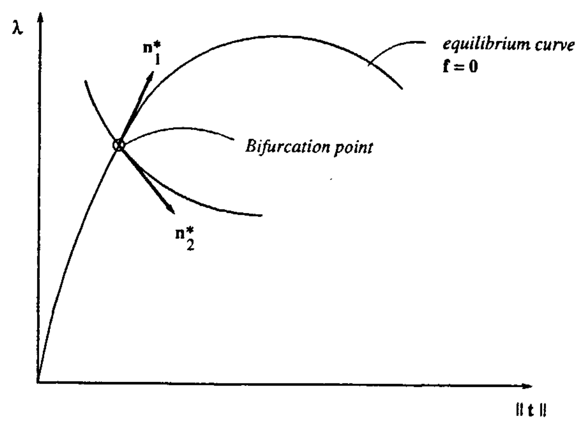


Figure 3.7: Bifurcation point

Consequently, no *unique* solution for the tangent and other path derivatives can be found at a bifurcation point. In practice, several methods have been developed to follow the intersecting solution curve.(See refs. [14] and [15]) for details).

3.2.5 Prescribed displacements

It was mentioned in chapter 2.4 that the external loading on a body can be prescribed by either force or displacement. For the method of prescribed displacement the interesting issue arises of how to determine the vector $\mathbf{f}_{,\lambda}$ in eqs. (3.11). The vector that represents the configuration space $\mathbf{t}^* \in \mathbb{R}_{N+m}$ can be written as:

$$\mathbf{t}^* = \begin{bmatrix} \mathbf{t} \\ \mathbf{t}_d \end{bmatrix} \qquad , \mathbf{t}_d = \lambda \mathbf{g} \tag{3.17}$$

with

$$\begin{aligned} \mathbf{t}^* &\in \mathbb{R}_{N+m} \\ \mathbf{t} &\in \mathbb{R}_N \\ \mathbf{t}_d &\in \mathbb{R}_m \end{aligned} \tag{3.18}$$

In (3.17) $\mathbf{t}$ represents the vector of non-prescribed displacements or computational degrees of freedom and $\mathbf{t}_d$ represents the vector of prescribed displacements.

The first of eqs. (3.11) can now be written as:

$$\mathbf{f}_{,\mathbf{t}}(\mathbf{t}^*, \lambda)\dot{\mathbf{t}} + \mathbf{f}_{,\lambda}(\mathbf{t}^*, \lambda)\dot{\lambda} = \mathbf{0} \tag{3.19}$$

with

$$\mathbf{f}_{,\lambda} = \frac{\partial \mathbf{f}(\mathbf{t}^*, \lambda)}{\partial \mathbf{t}_d} \frac{\partial \mathbf{t}_d}{\partial \lambda} + \frac{\partial \mathbf{F}(\lambda)}{\partial \lambda} \tag{3.20}$$

yielding

$$\mathbf{f}_{,\lambda} = \mathbf{f}_{,\mathbf{t}_d}(\mathbf{t}^*, \lambda)\mathbf{g} + \mathbf{F}_0 \tag{3.21}$$

In eq. (3.21) the component $\mathbf{f}_{,\mathbf{t}}(\mathbf{t}^*, \lambda)$ can be treated as the stiffness matrix $\mathbf{f}_{,\mathbf{t}}(\mathbf{t}, \lambda)$ in eqs. (3.11) and the term $\mathbf{f}_{,\mathbf{t}_d}(\mathbf{t}^*, \lambda)\mathbf{g} + \mathbf{F}_0$ can be considered as the vector $\mathbf{f}_{,\lambda}$ in eqs. (3.11). Notice the change in the stiffness matrix for prescribed displacements.

3.3 B2CONT routine

The *path following technique* as described in the previous section is available as the macro-processor **B2CONT**, which performs nonlinear static analysis by calling a sequence of processors. An explanation of these processors can be found in ref. [9].

The prediction method, correction method, auxiliary surface etc, used for the analysis can be selected by specifying some optional *PCL-commands* (PCL= Processor Command Language), some of which will be described below.

auxiliary surface
The choice for the auxiliary surface can be specified by the *PCL* **pathpar** as follows:

pathpar=0:
auxiliary surface parallel to the deformation axis, as described in 3.2.2, part *(i)*.

pathpar=1, 2:
auxiliary surface adjusted to the direction of the solution curve, using the unit tangent $\mathbf{n}^*$ to the solution curve.

actual prediction method that is used
The prediction method can be specified by the *PCL* command **extrapolate**. Depending on the specified value the computation is executed by Euler prediction, linear or quadratic extrapolation.

As long as one point on the solution curve is given, the first step will always make use of an Euler prediction. The known point can either be the origin or a solution point obtained from a previous computation (restart position). The subsequent prediction steps will be performed in the following sequence:

extrapolate= 0 (linear interpolation)
As soon as two solution points are available, i.e. at the subsequent loadsteps, the prediction is performed by linear extrapolation as requested.

extrapolate= 1 (quadratic extrapolation 1)
As quadratic extrapolation requires three available solution points, the prediction is continued by Euler-step prediction until three solution points are known. As soon as three solution points are known, the subsequent steps of the computation are made by quadratic extrapolation (method 1) as requested. The quadratic terms are ignored by this extrapolation method [15].

extrapolate = 2 (quadratic extrapolation 2:default)
Similar to **extrapolation**= 1, the prediction will be performed by Euler-step prediction until three solution points are known. At the next step, when three solution points are available, the subsequent predictions will be made by quadratic extrapolation (method 2) as requested.

extrapolate = 3 (Euler prediction)
All predictions are performed by Euler-step predictions as specified.

step-size control
A proper control of the step-size $\Delta\eta = \eta - \eta^k$ belongs to one of the crucial aspects of continuation techniques. It is determinating for the rate of convergence of the predictions and in fact essential to acquire an optimal usage of the (available) computation time. The choice for step-size control as provided by the **B2CONT** routine is based on the empirical formula:

$$\Delta\eta = \left(\frac{1}{2}\right)^{\alpha} \left(\eta^k - \eta^{k-1}\right) \quad with \;\; \alpha = \frac{I_k - 5}{4} \tag{3.22}$$

Basically, this means that if more than 5 iterations are required for convergence, the step-size will be reduced and if less than 5 iterations are required for convergence, the step-size is increased. However, if the step-size becomes too large and causes divergence, the step-size will be reduced to half its size and a new prediction will be made. This reduction of step-size due to convergence is not done at the first step. Consequently, the user must adjust the step-size if divergence occurs at the first step.

The load-step can be specified by the user in the **Analysis Directives Table** of the input-file (**adir**), by the parameter **dpas** for **loadcase a** and **dpbs** for **loadcase b**. By giving the (optional) PCL command *even*, a constant step-size will be used if allowed by the convergence criteria. A more detailed description of the required input parameters and available options provided for user-specification can be found in ref. [9].

Chapter 4

Finite element formulation

In the previous chapter a short description was given of *the path following technique* and the corresponding nonlinear solver **B2CONT** available in the **B2000** master code. For implementation of an element into the code the expressions for the first and second variation are required, representing respectively the (nonlinear) equilibrium equations and the tangential stiffness matrix. Consequently, the derivation of both the first and second variation of the potential energy of a cable element will be treated in sections 4.5 and 4.6. For this purpose use is made of the principle of stationary value of the potential energy. Prior to determining these variations, the expressions are transformed to the discretized finite element formulation.

4.1 Potential Energy Method

The total potential energy of a body generally consists of two parts; the internal (elastic) potential energy or strain energy U and the external potential energy V:

$$P = U + V \tag{4.1}$$

The strain energy U can be represented by the following expression:

$$U = \int \Phi(\varepsilon_x, ..., \gamma_{xy}, ...)dV = \frac{1}{2}\int (\sigma_x\varepsilon_x + ... + \tau_{xy}\gamma_{xy} + ...)dV \tag{4.2}$$

The strains are derivable from the displacement field $\mathbf{u}$.

The second part of the total potential energy is the potential energy of the external loads. The potential energy *(V)* of a given external force (i.e. with given magnitude and direction) is determined by the opposite of the work W performed by this force. By the theory of virtual work one can write:

$$\delta W = \frac{\partial W}{\partial \mathbf{u}} \cdot \delta\mathbf{u} = \mathbf{f}_{ext}^T \delta\mathbf{u} \qquad , \mathbf{u} = \mathbf{x} - \mathbf{x}_0 \tag{4.3}$$

with

$\mathbf{x}_0 =$ original position

$\mathbf{x} =$ (current) position vector continuous case

$\mathbf{f}_{ext} =$ the vector of external forces

Any positive work done by the external force establishes a deformation of the body. If the external force is taken away, the body strifes to resume it's initial natural (low potential energy) position. Accordingly, a positive work corresponds to a negative change of external potential energy and vice-versa. Hence, the external potential energy V is expressed by:

$$\delta V = -\delta W = -\mathbf{f}_{ext}^T \delta \mathbf{u} \tag{4.4}$$

With the external forces $\mathbf{f}_{ext}$ consisting of volume forces ($\mathbf{f}$) and surface forces ($\mathbf{p}$), the external potential energy can be expressed by:

$$V = -\int_V (\mathbf{f}^T \mathbf{u}) dV - \int_S (\mathbf{p}^T \mathbf{u}) dS \tag{4.5}$$

The total potential energy of an elastic body submitted to a conservative load system can thus be written as a function of the displacement function $\mathbf{u}$ and its derivatives with respect to the spatial coordinates, $x, y, ..$:

$$P = P(\mathbf{u}, \frac{\partial \mathbf{u}}{\partial x}, ...) \tag{4.6}$$

In these expressions, $\mathbf{u}$ represents the unknown (displacement) function, which is a continuous function of the position vector $\mathbf{x}$. By the theory of ref. [17] the total potential energy P is stationary.

Consequently, according to the basic theory equilibrium is governed by a stationary value of the total potential energy:

$$\delta P = \frac{\partial P}{\partial \mathbf{u}} \cdot \delta \mathbf{u} = 0 \tag{4.7}$$

Whenever such a variational principle exists, approximation functions can be defined, such as is done by the finite element method. This involves discretization of the continuous structure into a number of finite elements and consequently, a discretization of the displacements and the external forces. This is established through the definition of piecewise continuous displacement functions ($\mathbf{N}(\mathbf{x})$) as an approximation to the actual continuous deformation:

$$\mathbf{u} = \mathbf{N}(\mathbf{x})\mathbf{t} \tag{4.8}$$

with $\mathbf{t}$ the unknown discretized displacements or generalized coordinates. Consequently, the potential energy can be expressed in terms of $\mathbf{t}$ and the load intensity parameter λ, see also chapter 3:

$$P = P(\mathbf{t}; \lambda) \tag{4.9}$$

After discretization, the equilibrium equations of the reduced form (4.9) can be determined from eqs. (4.7) as follows:

$$f_i(\mathbf{t}, \lambda) = \frac{\partial P(\mathbf{t}, \lambda)}{\partial t_i}; \qquad i = 1, 2, ...N \tag{4.10}$$

Substitution of (4.8) into (4.2) and (4.5) and subsequently into (4.1) and (4.10) yields:

$$f_i(\mathbf{t}, \lambda) = \int \frac{\partial \Phi(\mathbf{t}, \lambda)}{\partial t_i} dV + \frac{\partial V}{\partial t_i} \tag{4.11}$$

As the external forces $\mathbf{f}_{ext}$ are assumed not to depend on the deformations of the structure, $\mathbf{t}$, eq. (4.11) yields:

$$\mathbf{f}(\mathbf{t}, \lambda) = \mathbf{f}_{int}(\mathbf{t}, \lambda) - \mathbf{f}_{ext} \tag{4.12}$$

Where $\mathbf{f}_{int}$ represents the vector of internal forces. Note that the equilibrium equations depend nonlinearly on the unknowns $\mathbf{t}$ and λ. Recall also section 2.1 and chapter 3.

Next, the stiffness matrix is required in order to enable to solve the problem by the *path following technique* as was discussed in chapter 3. Contrary to the stiffness matrix used for solving linear problems, this stiffness matrix is not constant but is dependent on the current configuration, $\mathbf{t}$. The tangential stiffness matrix can be obtained by differentiation of the equilibrium equations (4.12) with respect to the discretized displacements $\mathbf{t}$. The stiffness matrix is useful in determining whether a solution $\mathbf{t}$ is stable or not. According to the theory of stability of conservative systems a system is stable if the quadratic form:

$$P_2 = \frac{1}{2} \frac{\partial^2 P(\mathbf{t}; \lambda)}{\partial t_i \partial t_j} \Delta t_i \Delta t_j = \frac{1}{2} f_{i,j} \Delta t_i \Delta t_j \qquad i, j = 1, 2, ...N \tag{4.13}$$

is positive definite. If (4.13) is semi positive definite, the configuration $\mathbf{t}$ is in a critical state, else it is in an unstable state. The tangential stiffness matrix is now identified as:

$$\mathbf{K}_T = \frac{\partial^2 P(\mathbf{t}; \lambda)}{\partial t_i \partial t_j} = \frac{\partial \mathbf{f}(\mathbf{t}, \lambda)}{\partial t_i} = \mathbf{f}_{,\mathbf{t}} \qquad i, j = 1, 2, ...N \tag{4.14}$$

In the cases discussed here, the stiffness matrix is a symmetric matrix:

$$\mathbf{K}_T = \frac{\partial^2 P(\mathbf{t};\lambda)}{\partial t_i \partial t_j} = \frac{\partial^2 P(\mathbf{t};\lambda)}{\partial t_j \partial t_i} \qquad i,j = 1,2,...N \tag{4.15}$$

which is a useful property in further derivations.

4.2 Strain definition

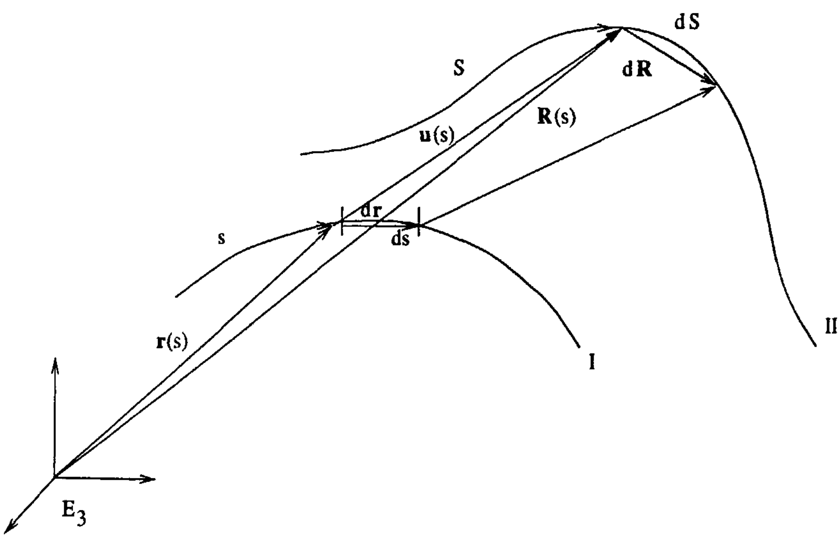


Figure 4.1: Strain definition

Before working out the expression for the potential energy we require an expression for the strain of a one dimensional body, i.e with only tangential (tension) stiffness. Figure 4.1 illustrates the deformation of a flexible cable from an arbitrary reference position (I) into a deformed position (II), where

$\mathbf{r}$(s) represents the position vector of the undeformed state

$\mathbf{R}$(s) represents the position vector of the deformed state

$\mathbf{u}$(s) represents the deformation or displacement vector

s represents the arc length along the undeformed cable

S represents the arc length along the deformed cable.

Note that the position vectors are defined with respect to the global reference system, E_3. From the figure we have:

$$\mathbf{R}(s) = \mathbf{u}(s) + \mathbf{r}(s) \tag{4.16}$$

Hence

$$\frac{\partial \mathbf{R}(s)}{\partial s} = \frac{\partial \mathbf{r}(s)}{\partial s} + \frac{\partial \mathbf{u}(s)}{\partial s} \tag{4.17}$$

where

$$\mathbf{r}(s) = [\, x(s)\mathbf{e}_1 + y(s)\mathbf{e}_2 + z(s)\mathbf{e}_3]^T, \qquad \mathbf{e}_i \in E_3 \tag{4.18}$$

The displacement vector $\mathbf{u}$(s) is assumed to be a *smooth* function, which means that $\mathbf{u}$(s) can be *differentiated* with respect to the arc length s. The position vector $\mathbf{R}$(s) depends on the arc length, s, but can also depend on additional parameters like time, load or space. Note that in accordance with the Lagrangean description, both position vectors $\mathbf{R}$(s) and $\mathbf{r}$(s) depend on the arc length s along the *undeformed* cable.

As a measure of deformation (or elongation), the most commonly used definitions for strain are the Engineering strain and Green-Lagrange strain, both according to the Lagrangean description. The Engineering strain ε_e is defined as

$$\varepsilon_e = \frac{dS - ds}{ds} \tag{4.19}$$

and the Green-Lagrange definition as:

$$\varepsilon_{gl} = \frac{1}{2}\frac{(dS)^2 - (ds)^2}{(ds)^2} \tag{4.20}$$

It is only for large strains that the two strain measures yield different results. Large strains inevitably involve additional complications, e.g. plastic deformations and will not be considered in this report. For small strains it is not relevant which strain definition is used.

In the case of implementing a cable element with string properties, the mathematical implications are limited to a minimum, as we are in fact dealing with a one dimensional problem.

The first and second variation in the present element routine are based on the Engineering strain definition, so that this definition will be used throughout this report. Figure 4.2 represents a partial enlargement of the deformed state in figure 4.1. Taking dS infinitesimally small, the arc length dS can be approximated by the length of the incremental position vector, $d\mathbf{R}$. Hence:

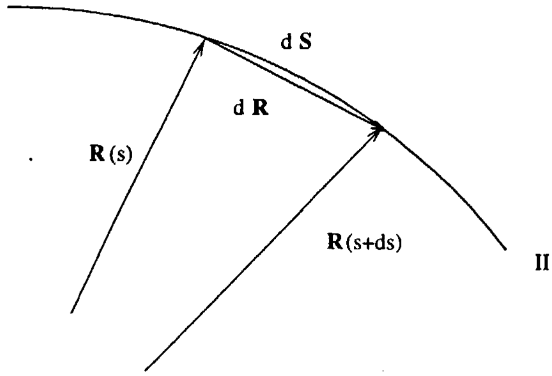


Figure 4.2: Unit tangent

$$(dS)^2 = d\mathbf{R} \cdot d\mathbf{R}$$

Similarly, for the undeformed state

$$(ds)^2 = d\mathbf{r} \cdot d\mathbf{r}$$

Taking the partial derivatives to the undeformed arc length s and rewriting yields:

$$(dS)^2 = (\frac{\partial \mathbf{R}}{\partial s} \cdot \frac{\partial \mathbf{R}}{\partial s})(ds)^2 \tag{4.21}$$

and

$$(ds)^2 = (\frac{\partial \mathbf{r}}{\partial s} \cdot \frac{\partial \mathbf{r}}{\partial s})(ds)^2 \tag{4.22}$$

Clearly, (4.22) is only true if $\mathbf{r}(s)$ obeys the following rule

$$\frac{\partial \mathbf{r}}{\partial s} \cdot \frac{\partial \mathbf{r}}{\partial s} = 1 \tag{4.23}$$

Any vector for which this property holds is called a *unit tangent*, in this report represented by $\mathbf{n}$;

$$\frac{\partial \mathbf{r}}{\partial s} = \mathbf{n}(s) \tag{4.24}$$

Rewriting (4.21) yields:

$$\frac{(dS)^2}{(ds)^2} = \frac{\partial \mathbf{R}}{\partial s} \cdot \frac{\partial \mathbf{R}}{\partial s} \tag{4.25}$$

Substitution of (4.25) into the Engineering strain definition (4.19) yields:

$$\varepsilon_e = \sqrt{\frac{\partial \mathbf{R}}{\partial s} \cdot \frac{\partial \mathbf{R}}{\partial s}} - 1 \tag{4.26}$$

Substituting (4.17) yields:

$$\varepsilon_e = \sqrt{(\frac{\partial \mathbf{r}}{\partial s} + \frac{\partial \mathbf{u}}{\partial s}) \cdot (\frac{\partial \mathbf{r}}{\partial s} + \frac{\partial \mathbf{u}}{\partial s})} - 1 \tag{4.27}$$

Finally, working out the inner-product and substitution of eqs. (4.23) and (4.24) yields:

$$\varepsilon_e = \sqrt{1 + 2\mathbf{n}(s) \cdot \frac{\partial \mathbf{u}}{\partial s} + \frac{\partial \mathbf{u}}{\partial s} \cdot \frac{\partial \mathbf{u}}{\partial s}} - 1 \tag{4.28}$$

4.3 Total Potential Energy of a cable element

For a perfectly flexible, one dimensional element like a cable (string) element with linear material properties, possessing only tension stiffness and thus incapable of resisting compression, shear or bending forces, the total potential energy is represented by:

$$P = \frac{1}{2} \int_V E\varepsilon^2 dV - W \qquad \varepsilon > 0 \tag{4.29}$$

Where W represents the work done by the external forces, if present. E represents Young's modulus, ε the strain and V the total volume of the body.

Implementation into a finite element software program requires the discretized equilibrium equations. Before substituting the expression for the strain ε_e , eq. (4.28), and working out the first and second variation, the total potential energy expression will be discretized.

4.4 Discretization

This section deals with the applied discretization, required to allow direct implementation of the discretized equilibrium equations and the tangential stiffness matrix into the **B2000** code.

The first step in discretization is to mesh the continuous cable element into n elements, see figure 4.3. Each element is considered separately to determine its tangential stiffness matrix and the equilibrium equations. To obtain the expressions for the complete structure, the contributions of each element are assembled by **B2000** in the conventional manner, see also ref. [12].

Consider a single straight two-node cable element with corresponding displacements as presented in figure 4.4, placed in the global reference system in

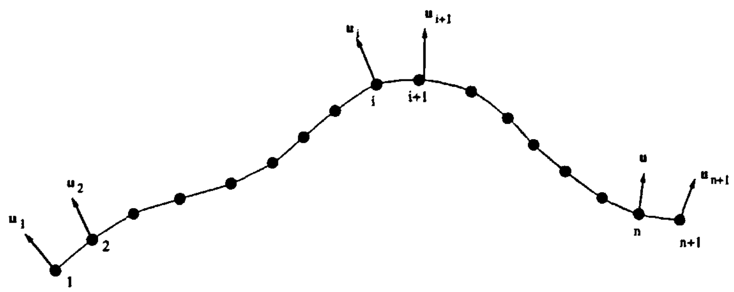


Figure 4.3: Discretization of a cable

an arbitrary direction. The deformed element is also considered to be straight. Obviously, many other options for the deformed shape are possible, but considering a straight element offers a simple mathematical derivation and direct implementation into the code, without essentially affecting the reliability or accuracy of the results.

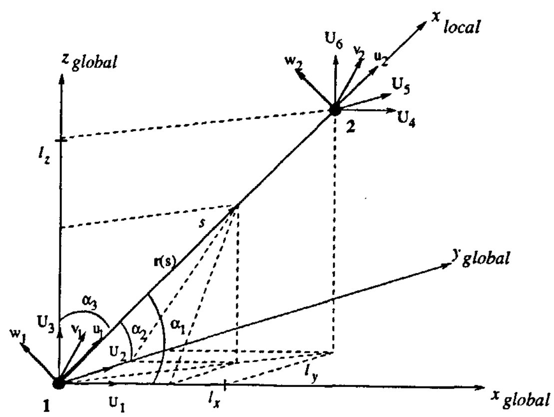


Figure 4.4: Discretization of one cable element

The global displacements are represented by U_i and the local displacements by u, v, w. The projections of the cable length on the global axes are subsequently l_x, l_y and l_z. Furthermore the initial length of the cable element is l_0 and the deformed length is l. The position vector $\mathbf{r}(s)$ can be expressed in terms of the position angles α_1, α_2 and α_3:

$$\mathbf{r} = \begin{pmatrix} s \cos \alpha_1 \\ s \cos \alpha_2 \\ s \cos \alpha_3 \end{pmatrix}_{global} = s \begin{pmatrix} \frac{l_x}{l_0} \\ \frac{l_y}{l_0} \\ \frac{l_z}{l_0} \end{pmatrix}_{global} \tag{4.30}$$

where

$$\begin{aligned} l_x &= x_2 - x_1 \\ l_y &= y_2 - y_1 \\ l_z &= z_2 - z_1 \\ l_0 &= \sqrt{l_x^2 + l_y^2 + l_z^2} \end{aligned} \tag{4.31}$$

The deformation of the cable element in the directions of the local reference system is approximated by a linear shape or interpolation function. This function depends on the position parameter of the undeformed cable s, see figure 4.5, and expresses the deformation in terms of the nodal displacements.

Conventionally, the displacement functions are defined with respect to the local element system and expressed in terms of the local nodal displacements $\mathbf{a}$. The linear interpolation functions of an element i are determined from:

$$\mathbf{u}_i^{(i)}(s) = (1 - \frac{s}{l_0})\mathbf{a}_i^{(i)} + \frac{s}{l_0}\mathbf{a}_{i+1}^{(i)} \tag{4.32}$$

where

$$\mathbf{u}^{(i)}(s) = \begin{pmatrix} u(s) \\ v(s) \\ w(s) \end{pmatrix}^{(i)} \quad \text{and} \quad \mathbf{a}^{(i)} = \begin{pmatrix} \mathbf{a}_i \\ \mathbf{a}_{i+1} \end{pmatrix}^{(i)} \tag{4.33}$$

with

$$\mathbf{a}_i^{(i)}(s) = \begin{pmatrix} u_i(s) \\ v_i(s) \\ w_i(s) \end{pmatrix}^{(i)} \quad , \quad \mathbf{a}_{i+1}^{(i)}(s) = \begin{pmatrix} u_{i+1}(s) \\ v_{i+1}(s) \\ w_{i+1}(s) \end{pmatrix}^{(i)} \tag{4.34}$$

yielding

$$\mathbf{u}^{(i)}(s) = \begin{pmatrix} u(s) \\ v(s) \\ w(s) \end{pmatrix}^{(i)} = \begin{pmatrix} 1-\zeta & 0 & 0 & \zeta & 0 & 0 \\ 0 & 1-\zeta & 0 & 0 & \zeta & 0 \\ 0 & 0 & 1-\zeta & 0 & 0 & \zeta \end{pmatrix} \begin{pmatrix} u_i \\ v_i \\ w_i \\ u_{i+1} \\ v_{i+1} \\ w_{i+1} \end{pmatrix}^{(i)} \tag{4.35}$$

with

$$\zeta = \frac{s}{l_0}$$

or, in matrix notation:

$$\mathbf{u}^{(i)}(s) = \mathbf{N}(s)\mathbf{a}^{(i)} \tag{4.36}$$

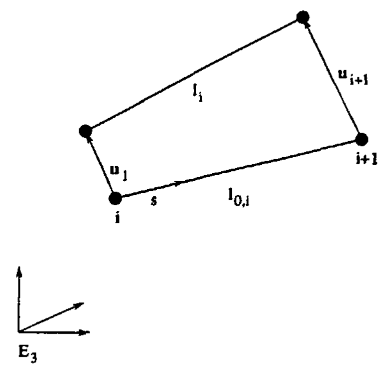


Figure 4.5: Linear interpolation of the deformed cable element

However, as we are dealing with a straight, two-node, one-dimensional element with linear displacement functions, it is obvious that these local displacements **a** can be directly transformed to their global equivalents **U**, while conserving the same interpolation functions. The interpolation functions can then be directly determined from:

$$\mathbf{U}_i^{(i)}(s) = (1 - \frac{s}{l_0})\mathbf{U}_i^{(i)} + \frac{s}{l_0}\mathbf{U}_{i+1}^{(i)} \tag{4.37}$$

where

$$\mathbf{U}^{(i)} = \begin{pmatrix} U_i \\ U_{i+1} \end{pmatrix}^{(i)} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{pmatrix}^{(i)} \quad \text{and} \quad \mathbf{U}^{(i)}(s) = \begin{pmatrix} U(s) \\ V(s) \\ W(s) \end{pmatrix}^{(i)} \tag{4.38}$$

yielding:

$$\mathbf{U}^{(i)}(s) = \begin{pmatrix} U(s) \\ V(s) \\ W(s) \end{pmatrix}^{(i)} = \begin{pmatrix} 1-\zeta & 0 & 0 & \zeta & 0 & 0 \\ 0 & 1-\zeta & 0 & 0 & \zeta & 0 \\ 0 & 0 & 1-\zeta & 0 & 0 & \zeta \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{pmatrix}^{(i)} \tag{4.39}$$

or, in matrix notation:

$$\mathbf{U}^{(i)}(s) = \mathbf{N}(s)\mathbf{U}^{(i)} \tag{4.40}$$

Having applied the above described discretization, the strain expression, eq. (4.28) can be rewritten in a discretized form;

$$\varepsilon_e = \sqrt{1 + 2\mathbf{r}'^T\mathbf{N}'\mathbf{U} + \mathbf{U}^T\mathbf{N}'^T\mathbf{N}'\mathbf{U}} - 1 \tag{4.41}$$

Where $'$ indicates (partial) differentiation with respect to the position parameter s. Consequently,

$$\mathbf{r}' = \begin{pmatrix} \cos\alpha_1 \\ \cos\alpha_2 \\ \cos\alpha_3 \end{pmatrix} = \begin{pmatrix} \frac{l_x}{l_0} \\ \frac{l_y}{l_0} \\ \frac{l_z}{l_0} \end{pmatrix} \tag{4.42}$$

and

$$\mathbf{N}' = \frac{1}{l_0}\begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix} \tag{4.43}$$

Note that for convenience the super-index $^{(i)}$ indicating the element id. is omitted as will be done in the sequel of this report. Substituting eqs. (4.42) and (4.43) into eq. (4.41) and working out the details, yields an expression for the strain in terms of the discretized global displacements $\mathbf{U}$:

$$\begin{aligned} \varepsilon_e = & \sqrt{1 + \frac{2}{l_0^2}\left[l_x(U_4 - U_1) + l_y(U_5 - U_2) + l_z(U_6 - U_3)\right]} \\ & \overline{+\frac{1}{l_0^2}\left[(U_4 - U_1)^2 + (U_5 - U_2)^2 + (U_6 - U_3)^2\right]} - 1 \end{aligned} \tag{4.44}$$

Note that for the case of a straight cable element as discussed in this section the position parameter s corresponds with the local x-axis along the undeformed cable, as can be clearly seen in figure 4.4.

4.5 First variation

Recall from section 4.3, eq. (4.29), that the total potential energy for a cable element is given by

$$P = \frac{1}{2}\int_V E\varepsilon^2 dV - W$$

Taking the first variation of the potential energy, according to eqs. (4.11) and (4.12) of section 4.1, with respect to the discretized displacements $\mathbf{U}$, yields

$$\mathbf{f} = \frac{\partial P}{\partial \mathbf{U}} = \int_{l_0} E\varepsilon \frac{\partial \varepsilon}{\partial \mathbf{U}} A ds - \frac{\partial W}{\partial \mathbf{U}} = 0 \tag{4.45}$$

with

$$\frac{\partial W}{\partial \mathbf{U}} = \mathbf{f}_{ext} \tag{4.46}$$

where $\mathbf{f}_{ext}$ represents the vector of external forces.
Hence, the nonlinear equilibrium equations for the cable with only axial (tension) stiffness are now represented by:

$$\int_{l_0} E\varepsilon \frac{\partial \varepsilon}{\partial \mathbf{U}} A ds - \mathbf{f}_{ext} = 0 \tag{4.47}$$

Implementation of the element routine within the **B2000** code requires only the first term of the equilibrium equations, representing the vector of internal forces $\mathbf{f}_{int}$. Consistent to the terminology used in **B2000**, the first term of the equilibrium equations shall be referred to as the first variation in the sequel. The vector of external forces is directly constructed by **B2000** from the defined load-cases in the input-file.

As the modulus of elasticity ,E, and the cross sectional area, A, were both assumed constant and also the strain is independent of s (see eq. (4.44)), the integrand is clearly not a function of s and can be computed separately as

$$\int_{l_0} ds = l_0 \tag{4.48}$$

resulting in

$$\mathbf{f}_{int} = EAl_0\varepsilon \frac{\partial \varepsilon}{\partial \mathbf{U}} \tag{4.49}$$

to represent the first variation. Recalling the discretized strain expression, (4.44) and differentiating with respect to the discretized displacements, yields the following derivatives

$$\frac{\partial \varepsilon}{\partial U_1} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(-l_x) + \frac{1}{l_0^2}(2(U_4 - U_1)(-1))\right) \tag{4.50}$$

$$\frac{\partial \varepsilon}{\partial U_2} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(-l_y) + \frac{1}{l_0^2}(2(U_5 - U_2)(-1))\right) \tag{4.51}$$

$$\frac{\partial \varepsilon}{\partial U_3} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(-l_z) + \frac{1}{l_0^2}(2(U_6 - U_3)(-1))\right) \tag{4.52}$$

$$\frac{\partial \varepsilon}{\partial U_4} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(l_x) + \frac{1}{l_0^2}(2(U_4 - U_1)(1))\right) \tag{4.53}$$

$$\frac{\partial \varepsilon}{\partial U_5} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(l_y) + \frac{1}{l_0^2}(2(U_5 - U_2)(1))\right) \tag{4.54}$$

$$\frac{\partial \varepsilon}{\partial U_6} = \frac{1}{2}(B)^{-\frac{1}{2}} \cdot \left(\frac{2}{l_0^2}(l_z) + \frac{1}{l_0^2}(2(U_6 - U_3)(1))\right) \tag{4.55}$$

where

$$B = 1 + \frac{2}{l_0^2}\left[l_x(U_4 - U_1) + l_y(U_5 - U_2) + l_z(U_6 - U_3)\right] + \frac{1}{l_0^2}\left[(U_4 - U_1)^2 + (U_5 - U_2)^2 + (U_6 - U_3)^2\right] \tag{4.56}$$

Back-substitution into eq. (4.49) yields:

$$\mathbf{f}_{int} = elfvar(i) = EA\varepsilon l_0 \begin{pmatrix} -(l_x + (U_4 - U_1)) \\ -(l_y + (U_5 - U_2)) \\ -(l_z + (U_6 - U_3)) \\ (l_x + (U_4 - U_1)) \\ (l_y + (U_5 - U_2)) \\ (l_z + (U_6 - U_3)) \end{pmatrix} \frac{1}{l_0^2\sqrt{B}} \tag{4.57}$$

Rewriting $\sqrt{B}$ as $\frac{l}{l_0}$ and working out the details, yields for the first variation vector

$$\begin{pmatrix} elfvar(1) \\ elfvar(2) \\ elfvar(3) \\ elfvar(4) \\ elfvar(5) \\ elfvar(6) \end{pmatrix} = \frac{EA\varepsilon}{l} \begin{pmatrix} -l_x^* \\ -l_y^* \\ -l_z^* \\ l_x^* \\ l_y^* \\ l_z^* \end{pmatrix} \tag{4.58}$$

where

$$\begin{aligned} l_x^* &= x_2 - x_1 + U_4 - U_1 \\ l_y^* &= y_2 - y_1 + U_5 - U_2 \\ l_z^* &= z_2 - z_1 + U_6 - U_3 \end{aligned} \tag{4.59}$$

represent the projections of the deformed cable length l:

$$l = \sqrt{(l_x^*)^2 + (l_y^*)^2 + (l_z^*)^2} \tag{4.60}$$

The above obtained first variation vector is as such implemented into the original element routine of *b2ep39.F.* (see app B). Note that the internal nodal reaction forces due to the stress in the cable are easily recognized in the first variation vector (*force* = *stress* · *cross sectional area*); decomposed in the global directions by the direction cosines, see also figure 4.6.

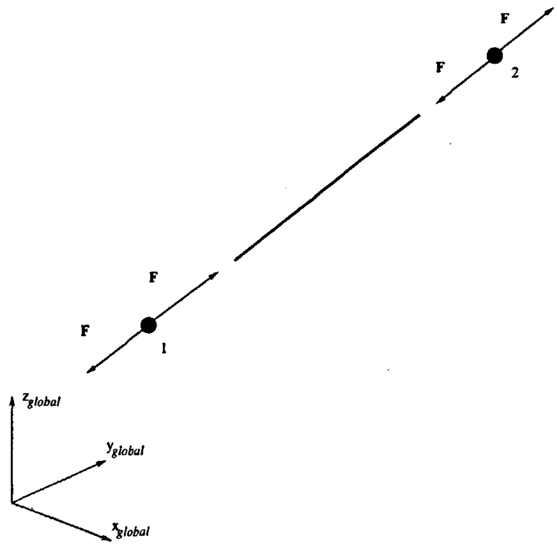


Figure 4.6: First variation

4.6 Second variation

As stated in the previous sections, the tangential stiffness matrix, K_T is represented by the second variation of the total potential energy, eq. (4.14).

The second variation is obtained by taking the derivatives of the components of the first variation vector with respect to each of the discretized displacements U_i, yielding a 6x6 tangential stiffness matrix in the case of a two node cable element placed in a 3-dimensional space and thus possessing a total of 6 degrees of freedom. Recall eq. (4.14). Hence, the tangential stiffness matrix is determined as follows:

$$\mathbf{K}_T = \begin{pmatrix} \frac{\partial F_1}{\partial U_1} & \frac{\partial F_1}{\partial U_2} & \cdots & \frac{\partial F_1}{\partial U_6} \\ \frac{\partial F_2}{\partial U_1} & \frac{\partial F_2}{\partial U_2} & \cdots & \frac{\partial F_2}{\partial U_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_6}{\partial U_1} & \frac{\partial F_6}{\partial U_2} & \cdots & \frac{\partial F_6}{\partial u_6} \end{pmatrix} \tag{4.61}$$

Where for the diagonal components:

$$\frac{\partial F_1}{\partial U_1} = \frac{\partial F_4}{\partial U_4} = \frac{EA}{l_0}\left(\left(\frac{l_x^*}{l}\right)^2 (1-\Delta) + \Delta\right) \tag{4.62}$$

$$\frac{\partial F_2}{\partial U_2} = \frac{\partial F_5}{\partial U_5} = \frac{EA}{l_0}\left(\left(\frac{l_y^*}{l}\right)^2 (1-\Delta) + \Delta\right) \tag{4.63}$$

$$\frac{\partial F_3}{\partial U_3} = \frac{\partial F_6}{\partial U_6} = \frac{EA}{l_0}\left(\left(\frac{l_z^*}{l}\right)^2 (1-\Delta) + \Delta\right) \tag{4.64}$$

and for the lower off-diagonal components:

$$\frac{\partial F_2}{\partial U_1} = -\frac{\partial F_5}{\partial U_1} = -\frac{\partial F_4}{\partial U_2} = \frac{\partial F_5}{\partial U_4} = \frac{EA}{l_0}\left(\left(\frac{l_x^*}{l}\right)\left(\frac{l_y^*}{l}\right)(1-\Delta)\right) \tag{4.65}$$

$$\frac{\partial F_3}{\partial U_1} = -\frac{\partial F_6}{\partial U_1} = -\frac{\partial F_4}{\partial U_3} = \frac{\partial F_6}{\partial U_4} = \frac{EA}{l_0}\left(\left(\frac{l_x^*}{l}\right)\left(\frac{l_z^*}{l}\right)(1-\Delta)\right) \tag{4.66}$$

$$\frac{\partial F_4}{\partial U_1} = -\frac{\partial F_1}{\partial U_1} \tag{4.67}$$

$$\frac{\partial F_3}{\partial U_2} = -\frac{\partial F_6}{\partial U_2} = -\frac{\partial F_5}{\partial U_3} = \frac{\partial F_6}{\partial U_5} = \frac{EA}{l_0}\left(\left(\frac{l_y^*}{l}\right)\left(\frac{l_z^*}{l}\right)(1-\Delta)\right) \tag{4.68}$$

$$\frac{\partial F_5}{\partial U_2} = -\frac{\partial F_2}{\partial U_2} \tag{4.69}$$

$$\frac{\partial F_6}{\partial U_3} = -\frac{\partial F_3}{\partial U_3} \tag{4.70}$$

where

$$\Delta = \frac{l - l_0}{l} \tag{4.71}$$

The components of the tangential stiffness matrix can also be written in index notation as follows:

$$\frac{\partial F_i}{\partial U_j} = \frac{EA}{l_0}\left(\left(\frac{l_i^*}{l}\right)^2 (1-\Delta) + \Delta\right) \quad i = j \;\; i,j = 1,2,3 \tag{4.72}$$

and

$$\frac{\partial F_i}{\partial U_j} = \frac{EA}{l_0}\left(\left(\frac{l_i^*}{l}\right)\left(\frac{l_j^*}{l}\right)(1-\Delta)\right) \quad i \neq j \;\; i,j = 1,2,3 \tag{4.73}$$

with

$$\frac{\partial F_i}{\partial U_j} = \frac{\partial F_{i+3}}{\partial U_{j+3}} = -\frac{\partial F_{i+3}}{\partial U_j} = -\frac{\partial F_i}{\partial U_{j+3}} \tag{4.74}$$

These components are as such implemented in the element routine *b2ep39.F*, see appendix B. Take notice of the fact that the matrix is *symmetric* and hence only the upper or lower triangle components need to be specified.

Summarizing, the following assumptions have been made in order to determine the first and second variation components:

- The stress-strain relationship obeys Hooke's law for linear elastic deformations, i.e. $\sigma = E\varepsilon$.
- The deformations remain within the linear elastic range during the whole nonlinear computation.
- The cross-section, A, and Young's modulus, E, are constant throughout the element and remain unchanged during the deformation process.
- Only conservative load systems are considered, see also section 2.1.
- The external loads are not a function of the displacements.
- Linear and smooth interpolation functions are used to describe the deformation of the (cable) element. Note that this is also the exact solution for a cable submitted to tension in quasi-static analysis.

Both the equilibrium equations and the tangential stiffness matrix are implemented into the element routine (*b2ep39.F*).

The expressions for the first and second variation derived in this chapter are valid for compression and tension states and can be used for the modeling of e.g. pin-jointed trusses. However, these expressions may only be used to describe the behaviour of the *cable* element for *tension* cases; i.e. $\sigma > 0$. This is due to the fact that a cable has zero compression stiffness and thus its behaviour under compression cannot be described by the same expressions as for tension.

Setting the stiffness matrix equal to zero for compression will inevitably yield a singular matrix and consequently the equilibrium equations cannot be solved for the unknown displacements. Alternative formulations for these required expressions must be found for the case of negative stress. See also chapters 5 and 6.

Chapter 5

Present formulation

It was stated in chapter 1 that the **B2000** software package already provides a nonlinear cable element. This chapter will discuss a selection of desired properties that are to be checked, modified (activated) and/or implemented into the existing element routine. This requires a proper evaluation of the possibilities offered by the present cable element. Section 5.1.2 gives an extensive evaluation of the present element routine.

The first elementary modifications to the element routine are discussed in section 5.2. These modifications involve corrections to guarantee correct implementation of the first and second variation definitions as derived in sections 4.5 and 4.6. The corrections only affect positive stress computations. Some example problems are treated to test the behaviour of the cable in positive stress situations. Recall section 2.4.

The negative stress states require a new approach to the definitions of the first and second variation. Aspects involving these singularity problems and possible options to approach these problems will be discussed in the next chapter (chapter 6).

5.1 Validation of existing element routine

5.1.1 Validation of b2epv39.F

The pre-variational element routine *b2epv39.F*, (see appendix A) computes the pre-variational data for a two-node cable element. These data define the initial geometry of the cable element and are required for computation of the first and second variation in *b2ep39.F*. As the pre-variational data are constant, *b2epv39.F* need only be called once at the start of the quasi-static computation. The final values of the pre-variational data are stacked in the array *elprev(i)* as follows:

elprev(1) = initial length in global x-direction, l_x

elprev(2) = initial length in global y-direction, l_y

elprev(3) = initial length in global z-direction, l_z

elprev(4) = initial (i.e. un-stressed) length, l_0

elprev(5) = initial cable cross sectional area, A

The current formulation does not offer any possibility to start the computation from a zero-stress configuration and thus a *prestress* must always be defined. Appendix C presents the algorithm part of the pre-variational routine which is essential for the *prestress* option. The node co-ordinates with respect to the global reference system are represented by $\mathbf{x}_1$ and $\mathbf{x}_2$ for respectively node 1 and node 2. The components of the length in the global directions are obtained by $\mathbf{x}_2 - \mathbf{x}_1$. The directive ecckern(1) is given a negative value to indicate prescribed initial length l_0 and a positive value to indicate prescribed *'prestress'*.

The deformed length l at zero load-increment, $l(0)$, will be referred to as l_0^* in the sequel of this report. This is the (initial) length which follows from the given node co-ordinates as defined by eq. (4.31) in section 4.4. Notice that for zero *prestress* the initial length l_0 equals the length at zero load-increment, l_0^*, yielding $l_0 = l_0^*$.

Prestress can be imposed by prescribing an initial length l_0 smaller than the length determined from the initial node positions, l_0^*, or by directly prescribing a *'prestress'* value. It must be remarked that in the input-file the parameter **prestress** must be given as a force, i.e. in Newtons, to be compatible with the definitions used in the original pre-variational element routine *b2epv39.F*. Only either value must be passed on to the element routine *b2ep39.F*, to allow an unambiguous algorithm definition of the first and second variation.

As the total stress and strain are expressed in terms of the initial length, l_0, and the deformed length, l, the most convenient choice for this purpose would be to use the initial length l_0. *b2epv39.F* computes the initial length for either prescribed initial length or prescribed initial stress and stores this value in the array component *elprev(4)*, ready to be read by the element routine *b2ep39.F*.

It must be remarked in this context that the different meanings of the word 'prestress' as used here, may lead to confusion. It has been attempted throughout the remainder of this report to limit this confusion as much as possible by using different fonts: In general the input parameters are printed in **boldface** type (e.g. **prestress, l0, mass**, etc) and variables in the element routine are printed in *italics* (e.g. *ae, itens, etc.*).

Notice that a rod element is implicitly defined in these routines and distinguished from the cable element by adding a minus sign to certain variables.

5.1.2 Validation of b2ep39.F

Similar to the pre-variational routine a flow-diagram is presented of the existing element routine, see appendix D. Contrary to *b2epv39.F*, which is called only once at the start of each quasi-static computation, the element routine *b2ep39.F* is called by **B2CONT** after each load-step. Recall also chapter 3. The performed analysis depends on the values of the directives *dirkern(1)*, *dirkern(2)* and *dirkern(3)*.

These directives direct the computation as follows:

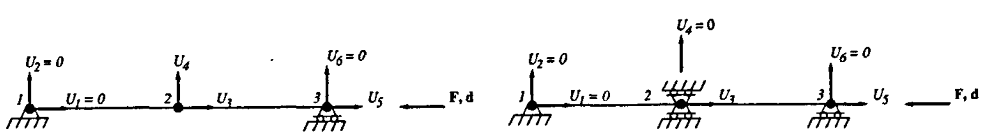
dirkern(1) : executes nonlinear analysis if set to 1 and
linear analysis if set to 0
dirkern(2) : computes first variation if set to 1
dirkern(3) : computes second variation if set to 1

First the items common to first and second variation are computed, e.g. l, l_0^* and **r'** (direction cosines; see eq. (4.42)). For linear analysis the pre-variational routine needs to be called explicitly, whereas for nonlinear analysis it is called implicitly by the **B2CONT** routine. If both directives *dirkern(3) and dirkern(1)* are set to 1, the element routine computes the second variation for the geometric nonlinear case, as specified in section 4.6. Before actually computing the second variation the element is checked for negative stress.

It was explained in chapter 2 that if the cable is in a state of tension it is in a stable condition. By reducing the tension to zero the cable becomes unstable. The cable can then take on any arbitrary configuration as long as it satisfies the kinematic condition of unchanged total arc length ($\varepsilon = 0$). The tensionless state sets the lower bound of possible conditions for the cable; a state of negative stress does simply not exist. Obviously, alternative formulations must be found for computations with $\varepsilon < 0$.

First, let us consider the tensionless state, i.e. the transition between tension and compression. The condition of the stiffness matrix for the tensionless case will be illustrated by two simple 2-dimensional examples. See figure 5.1. Both figures represent a simply supported cable modeled by two cable elements. In the left figure the mid-node is 'free' yielding a total of 3 (computational) degrees of freedom. In the right figure the lateral degree of freedom at the mid-node is removed.

In accordance with sections 2.2 and 2.3 the stiffness matrix of the problem presented in the left figure will be singular and in the right problem the stiffness matrix is definite as the degree of freedom causing the singularity is eliminated. These situations will be illustrated by considering the tangential stiffness matrix as derived in section 4.6.



(a) Singular stiffness matrix (b) Definite stiffness matrix

Figure 5.1: Simple cable problem to illustrate singularity at mid-node

At the instant the initial elongation is reduced to zero, the deformed length l becomes equal to the initial length l_0. Recalling the expressions in section 4.6, this means:

$$l = l_0$$

$$\varepsilon = 0$$

$$\Delta = \frac{l-l_0}{l} = 0$$

$$l_x^* = l_x$$

$$l_y^* = l_y$$

$$l_z^* = l_z$$

This yields the following assembled stiffness matrix (with the third degree of freedom at each node omitted), see eqs.(4.62) to (4.70):

$$\mathbf{K}_T = \frac{EA}{l_0} \begin{pmatrix} \gamma & 0 & -\gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\gamma & 0 & 2\gamma & 0 & -\gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma & 0 & \gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.1}$$

with

$$\gamma = \left(\frac{l_x}{l_0}\right)^2$$

Clearly, if only U_1, U_2 and U_6 are eliminated, (figure 5.1(a)) the stiffness matrix is singular for U_4. Eliminating U_4 as a freedom restores the regualrity. Note also that defining a prestress means that $l > l_0$, yielding non-zero Δ resulting in non-zero values at the diagonal of the stiffness matrix; the singularity is then eliminated.

The existing element routine establishes this by setting Δ equal to zero for the second variation computation. As negative stress conditions do not exist, the stress in the cable remains equal to zero for further compression ($\varepsilon < 0$). This is established in the element routine by treating the cable at zero stress for values $\varepsilon < 0$. This is conform the physical reality, but does not allow computation of situations involving tensionless states as described in section 2.4.

Additionally, the computation of the first variation in the existing element routine is continued with negative stress values. Obviously, this is intolerable for a cable with no bending stiffness and thus requires some modifications. The modifications required for the proper computation of the cable in tension is discussed in the following subsection. Modifications with respect to tensionless states requires a more extensive evaluation. This will be discussed in chapter 6.

5.2 Modification to present element routine

modifications with respect to a proper evaluation of the first variation
Recall the directives *dirkern(i)* described in section 5.1.2. These directives are alternately set to 1 and 0 in subsequent calls as indicated below.

dirkern(1)	1	1	1	1	1
dirkern(2)	1	0	1	0	1
dirkern(3)	1	1	0	1	0

Clearly, all directives are set to 1 only at the first call to *b2ep39.F.* This situation only occurs at the first call of **b2fact**, [9],[8], requesting computation of both first and second variation for non-linear analysis. Only at this call an error occurs in the computation of the first variation. See the flow-diagram in appendix D.

The stiffness parameter $ae = AE$ is assigned a new value $ae = \frac{AE}{l_0}$ prior to second variation computation. Although the second variation is computed correctly with this new value, *ae* has an incorrect value for first variation computation. In order to guarantee correct computation of the first variation if the directives *dirkern(3)* and *dirkern(2)* are both set equal to one the value of the stiffness parameter *ae* must be restored to $A \cdot E$.

The first variation in the origin is now computed correctly. In database terms: the data sets *VECTOR* **FVAR.GLOB.0** and **FVAR.i.0** are now correct (i indicates the load-cycle number). See refs. [9] and [8] for a more detailed declaration of these data sets.

modifications with respect to negative stress check
As can be seen from appendix D a negative stress check is performed only prior to second variation computation. Obviously, such a check is also required prior to first variation computation to avoid that the computation continues with negative values for the first variation in case of negative stress.

Some basic test examples which comprise testing the prestress option and the prescribed load/displacement options will be discused in section 5.3

5.3 Test examples

5.3.1 UN-symmetric triangular cable construction (testing present example problem)

In the *B2000 Processors Reference Manual*, see ref. [9], an example problem is given of static analysis of a simple cable construction. The construction consists of two branches; each consisting of two cable elements. The cables are given a prestress by prescribing the initial length l_0 ($< l_0^*$) of the elements. Due to the prestress in the cables a reaction force needs to be defined at the top-node, to guarantee equilibrium of the pre-stressed state. The external load on the construction is applied by superposition of an incremental load, or directly by incrementally increasing the value of the 'reaction' force by a given increment to a desired end value. The direct method can be applied anytime as long as the imposed external load has the same direction as the reaction force. A separate analysis is executed for the first three load increments after which a restart is executed to increase the load to a large value (1000 N in this example).

The main purpose of this test example is in the first place to regenerate the results obtained by the *Modified Newton Raphson Method* **B2MNR** in the *B2000 Processors reference manual* using the **B2CONT** solver with the element routine modified for tension. As stated in section 5.2, the correction to the stiffness parameter *ae* for the initial prediction is not expected to affect the numerical results much.

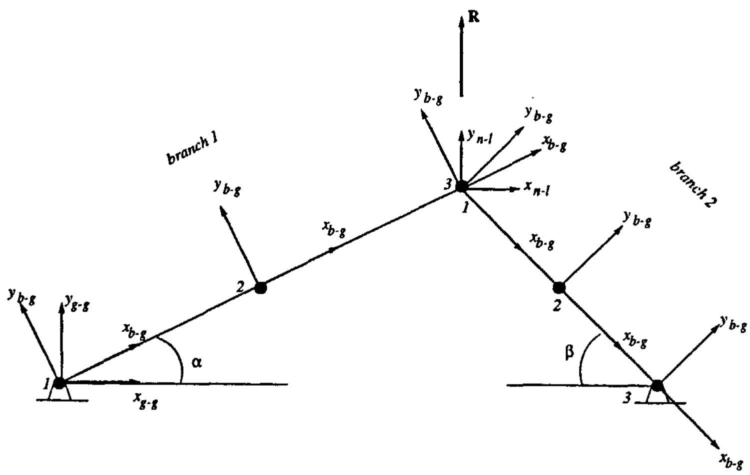


Figure 5.2: Geometry of un-symmetric cable construction

model definition
Figure 5.2 shows the geometry of the model, specifying the applied co-ordinate systems and the definition of the elements and nodes. The figure shows the *global-global* system (g-g), the *branch-global* systems (b-g) and one *node-local* system (n-l), [9] and [8]. The model data are given in table 5.1

	branch 1	*branch 2*
prestress	3.726779963 N	4.714045208 N
l	79.0450030 m	50.0 m
A	1.00 m^2	1.00 m^2
E	100.0 N/m^2	100.0 N/m^2
α	26.56505118^o	
β		45^o

Table 5.1: Model parameters

Starting with a given reaction force **R**= 5.0 $[N]$, the prestresses in the cables

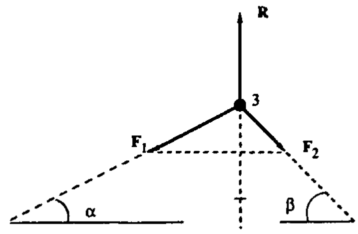


Figure 5.3: Reaction force due to prestress in cables

can be determined from figure 5.3 by solving the following set of equations for $\mathbf{F}_1$ and $\mathbf{F}_2$:

$$\begin{cases} \mathbf{F}_1 sin\alpha + \mathbf{F}_2 sin\beta = \mathbf{R} \\ \mathbf{F}_1 cos\alpha = \mathbf{F}_2 cos\beta \end{cases} \tag{5.2}$$

Substitution of the numerical values for α, β and $\mathbf{R}$, (see also table 5.1) yields the following cable forces:

$$\mathbf{F}_1 = 3.726779963\ N \tag{5.3}$$

$$\mathbf{F}_2 = 4.714045208\ N \tag{5.4}$$

As in this example the cross-sectional area of the cables is 1.0 $[m^2]$, the (pre-) stresses in the cables have the same values (see table 5.1). Next, the corresponding initial length can be computed from eq. (2.15):

$$l_0^i = \frac{l}{\frac{\mathbf{F}_i}{AE} + 1}$$

yielding for branch 1:

$$l_0^1 = \frac{79.0450030}{\frac{3.726779963}{100.0} + 1} = 76.20500996\ m \tag{5.5}$$

and for branch 2:

$$l_0^2 = \frac{50.0}{\frac{4.714045208}{100.0} + 1} = 47.74908648\ m \tag{5.6}$$

These model definition data are finally defined in an input-file (see ref. [9], *p. 144-149*). For completeness a more descriptive input-file is presented in appendix F, to offer a clearer picture of how a model can be defined by defining different branches and reference system rotations and translations.

results before and after restart

The results of the original model obtained by the *Modified Newton Raphson Method* are presented in table 5.2.

VECTOR ***DISP.GLOB.2***			
Node id.	**U [m]**	**V [m]**	**W [m]**
1	0.00000E+00	0.00000E+00	0.00000E+00
2	2.2209	9.8339	0.00000E+00
3	4.4416	19.668	0.00000E+00
Element id.	**cable force** F_i $[N]$	**stress** $\sigma[N/m^2]$	**strain** ϵ
1	8.896	8.896	8.8964E-02
2	8.896	8.896	8.8964E-02

Table 5.2: Results branch 1 (pa=0.15000E+02) according to example in *Processors Reference Manual*

VECTOR ***DISP.GLOB.2***			
Node id.	**U [m]**	**V [m]**	**W [m]**
1	0.00000E+00	0.00000E+00	0.00000E+00
2	2.2208	9.8339	0.00000E+00
3	4.4417	19.668	0.00000E+00
Element id.	**cable force** F_i $[N]$	**stress** $\sigma[N/m^2]$	**strain** ϵ
1	8.89687	8.89687	8.89687E-02
2	8.89687	8.89687	8.89687E-02

Table 5.3: Results branch 1 (pa=0.15000E+02) by **B2CONT**

The data set containing the global displacements (**DISP.GLOB.2**) of branch 1 at the final load-step (pa=15) are given. Also the internal cable forces $\mathbf{F_i}$, stresses σ and strains ϵ are given. The corresponding numerical results obtained from the **B2CONT** analysis are given in table 5.3.

Comparing the results presented in the tables 5.2 and 5.3 one can see that these coincide within an acceptable margin of approximately 0.0005%. Next, the computation is restarted from the last load-increment, i.e load increment 2. The force on the top-node is increased incrementally in this restart computation from **pa** =15.0 to **pa** = 1000 $[N]$. The script-file used for this purpose is given in appendix F.

The deformed structure is presented in figure 5.4 and the corresponding load-displacement diagram in figure 5.5. The displacements of node 3 (branch 1) in the x-direction ($u(3)$) and y-direction ($v(3)$) are given with respect to the branch-global system by respectively graph 1 and graph 2. (See also figure 5.2) As expected this diagram is near to identical to the load-displacement diagram obtained by *Modified Newton Raphson* analysis.

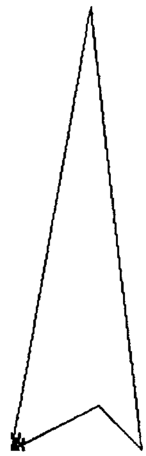


Figure 5.4: Deformed position TriaCable example

5.3.2 Simple example problem to test prestress option

model definition

For the purpose of testing the prestress option offered in the element routine a line cable model consisting of two cable elements is created. See figure 5.6. The cable elements are pre-stressed by different values of prestress, requiring reaction forces at nodes 2 and 3 to guarantee equilibrium of the pre-stressed state. As two cable elements with different prestress are connected to each other, an interaction between the two elements can be expected. The model data are presented in table 5.4 below. The pre-stresses can be computed directly from $\mathbf{F}_i$ by dividing by the cross-sectional area, A.

	element 1 (i=1)	*element 2 (i=2)*
$\mathbf{F}_{0_i}$	0.2 N	0.1 N
l_{0_i}	4.0 m	6.0 m
$l^*_{0_i}$	4.4 m	6.3 m
E	1.0 N/m^2	1.0 N/m^2
A	2.0 m^2	2.0 m^2

Table 5.4: Model data

The initial lengths and pre-stresses of the cable are given. Note that the cable data do not represent realistic values and are purely chosen to illustrate and test specific cable properties. By considering node equilibrium the reaction forces $\mathbf{R}_1$ and $\mathbf{R}_2$ can be determined. This yields:

$$\mathbf{R}_1 = 0.1\ N \qquad \mathbf{R}_2 = 0.1\ N \tag{5.7}$$

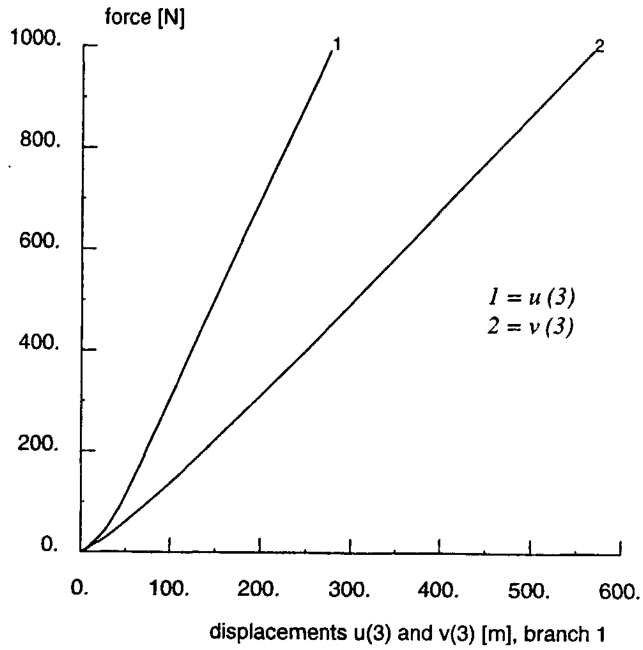


Figure 5.5: Load-displacement diagram of node 3 branch 1, TriaCable example

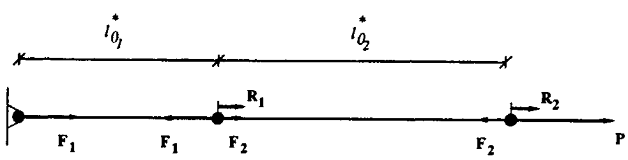


Figure 5.6: Simple line model to test prestress option

The deformed lengths can be determined from eq. (2.15), yielding for the prestressed lengths $l^*_{0_1} = 4.4\ m$ and $l^*_{0_2} = 6.3\ m$ respectively, see table 5.4. To test the pre-stress option, the computation is executed twice; once with prescribed 'initial stress' **prestress** and once with prescribed 'initial length', **l0**. The results must coincide and are tested analytically.

results prescribed 'prestress'
The input-file of the example problem described above is given in appendix F.

The reaction forces are defined in **loadcase b** and kept constant. An external force **P** (**loadcase a**) is superposed incrementally from 0.0 N (**pas**) to 0.4 $[N]$ (**pamax**). This yields internal cable forces of 0.6 $[N]$ ($\mathbf{F}^*_1$) for element 1 and 0.5 N ($\mathbf{F}^*_2$) for element 2, see also figure 5.6. Substituting the obtained and given values in eq. (2.15) and solving for the deformed length l yields:

$$l_1 = l_0 \left(\frac{\mathbf{F}_1^*}{EA} + 1 \right) = 5.2\ m \tag{5.8}$$

for element 1 and

$$l_2 = l_0 \left(\frac{\mathbf{F}_2^*}{EA} + 1 \right) = 7.5\ m \tag{5.9}$$

for element 2. The corresponding horizontal nodal displacements u with respect to the node positions defined in the input-file are given in table 5.5:

node id.	*displ. (u)*
2	0.8
3	2.0

Table 5.5: Model parameters

The obtained numerical results are given in table 5.6. As can be seen from table 5.6 the obtained displacements of the nodes 2 and 3 correspond with the analytical results in table 5.5. Note that the load is increased in two load steps.

*displacements **u** in branch-global*		
Node id.	**DISP.1.1 [m]**	**DISP.1.2 [m]**
1	0.0000E+00	0.0000E+00
2	4.0000E-01	8.0000E-01
3	1.0000E+00	2.0000E+00

Table 5.6: Nodal displacements per load step

results prescribed initial length l_0
Having tested the **prestress** parameter, remains to test the results obtained by prescribing the initial length l_0 (**l0** parameter).

Again executing quasi-static analysis with **B2CONT** does indeed yield results identical to the previously obtained results given in table 5.6. Clearly, the results correspond with the results obtained by prescribed **prestress**.

5.3.3 Simple example problem to test prescribed displacement option

The next element property we expect to work correctly after having activated the code for the existing cable element and having applied the *ae* correction to the element routine, is the *prescribed displacement* option. The relationship between prescribing displacements and prescribing forces was explained in sections 3.2.5 and 2.4. For identical problems the results obtained by prescribed displacements must coincide with the results obtained using prescribed forces (loads).

For this purpose use is made of the same model as was used in the previous subsection, see figure 5.6.

model definition; input file for prescribed displacements
Displacements are prescribed by using **type D** for the **force** parameter. This incremental (displacement) load is defined in **loadcase a**. The definition of the reaction forces remains unchanged. The end-value of the prescribed displacement is 2.0 $[m]$. This is the horizontal displacement u of node 3 computed in the previous subsection 5.3.2 for prescribed force 0.4 $[N]$. Executing the same problem with prescribed displacement should thus result in a (total)reaction force of magnitude 0.5 $[N]$ on node 3 and cable forces of 0.5 $[N]$ in cable 1 and 0.6 $[N]$ in cable element 2.

evaluation/validation of obtained results
The computed results are stored in datasets in the computational data base, **b**. The results of interest are presented in table 5.7.

VECTOR FVAR.1.i		
node id.	**FVAR.1.0 [N]**	**FVAR.1.2 [N]**
1	0.0000000E+00	0.0000000E+00
2	1.0000000E-01	1.0000000E-01
3	1.0000000E-01	0.5000000

Table 5.7: Results prescribed displacement

By considering node equilibrium the internal cable forces are found to be 0.6 $[N]$ for element 1 and 0.5 $[N]$ for element 2, corresponding to the findings in subsection 5.3.2. Consequently, the consistency of the prescribed displacement option is proven.

5.3.4 Simple cable sag problem

model definition
Recall from section 2.4 the case of a suspended cable with given geometry for the unloaded state. This subsection will test the method described in section 2.4 by a simple example problem.

Consider a cable with initial length $l_0 = 4.0$ $[m]$, submitted to a uniformly distributed transverse loading. The cable is discretized into 4 elements, each with equal lengths and properties. The load is discretized accordingly, see figure 5.7

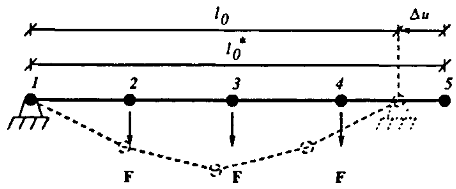


Figure 5.7: Simple cable example to illustrate cable sag computation by methods avoiding negative stress

The span of the requested deformed state is given and equal to the undeformed cable length l_0. However, as was pointed out in the previous sections, problems will arise at the initial step of the computation if the computation is started from an unloaded state. Therefore the cable is given a pre-stress with total cable length l_0^* of 4.4 $[m]$, corresponding to an initial stress of $\sigma_0 = 10^6$ $[N/m^2]$ or **'prestress'** (given in $[N]$) of 100 $[N]$. These and further data are presented in table 5.8.

	data per element or node
F	10.0 N
l_{span}	4.0 m
l_{0_i}	1.0 m
$l_{0_i}^*$	1.1 m
E	10^6 N/m^2
A	0.0001 m^2

Table 5.8: Model data for cable sag example

Note that the data in table 5.8 are again chosen for illustrative means. To obtain the desired end configuration the initial elongation Δu is reduced incrementally back to zero as the forces **F** are incrementally increased from zero to the prescribed end value (10.0 $[N]$). The pre-stressed state must be equilibrated by reaction forces equal to the value of **'prestress'** (100 [N]). However, as use is made of prescribed displacements the reaction forces will change each load increment and thus need not and may not be prescribed separately.

Both **F** and Δu can be prescribed within one loadcase, **loadcase a (lca)**, from start values **pas** with load-steps **dpas*loadfactor** to end values **pamax*loadfactor**. See appendix F for the corresponding input file.

validation of obtained results
The results obtained by **B2CONT** for the problem described above are presented in table 5.9:

node id.	*x-disp*	*y-disp*
1	0.0000E+00 m	0.0000E+00 m
2	-2.1316E-01 m	-8.3590E-01 m
3	-2.0000E-01 m	-1.1856E+00 m
4	-1.8684E-01 m	-8.3590E-01 m
5	-4.0000E-01 m	0.0000E+00 m

Table 5.9: Results cable sag example

The obtained final position corresponds to that of a cable suspended between two hinges a distance l_0 apart and submitted to a prescribed transverse loading. As mentioned in section 2.4 part *(v)* direct computation of the deformed state is desirable. Methods to enable this direct computation will be discussed in the next chapters. The results must of course correspond to the results obtained in this example.

Chapter 6

Approaches to singularity problem

In the previous chapter example problems were treated to illustrate the behaviour of the cable under prestress. This chapter treats the options that were considered to remove the singularity when the cable stresses become negative. The advantages and disadvantages of the various options are discussed briefly and illustrated by example problems where necessary.

The options discussed in the subsequent sections involve an interpretation of the negative stress conditions, an evaluation of mechanical formulation, implementation (modifications to the element routine) and finally an evaluation and explanation of the results. An approach which is physically correct does not necessarily lead to problem-free computations as will become clear in this chapter.

The approach that is finally implemented into the element routine involves an adjustment of both the initial length of the tensionless cable and the modulus of elasticity. Full details of this method are discussed in section 6.5

6.1 AE (=0) method

6.1.1 Solution procedure for AE-method

The most obvious and simple approach to negative stress is to eliminate the contribution of cables with negative stress to the stiffness of the total cable construction. This can be achieved by setting the stiffness parameter AE equal to zero for negative stress, resulting in zero contribution to the first and second variation and hence in fact eliminating the 'presence' of the concerned cables.

When the stress in the cables becomes positive again at some point of the deformation process the contribution of these cables to the total stiffness must of course be taken into account again. Recall also section 2.4.

6.1.2 Modifications to element routine

In case of negative stress the parameter **itens** is set to 1. This parameter thus functions as a directive for modifications to the first and second variation. For elements with negative stress the stiffness parameter *ae* is set equal to zero. If the cable element resumes its state of tension *ae* is restored to its original value.

Next, this method is tested on some example problems representing the negative stress situations described in section 2.4, parts *(iv)* and *(v)*.

6.1.3 Example problems

example problem 1: cable as part of a construction
The first property that will be tested is the behaviour of a (temporary) 'tensionless' cable as part of a construction as described in section 2.4, part *(vi)*. The cables are discretized into two cable elements each, yielding one interior-node per cable, see figure 6.1.

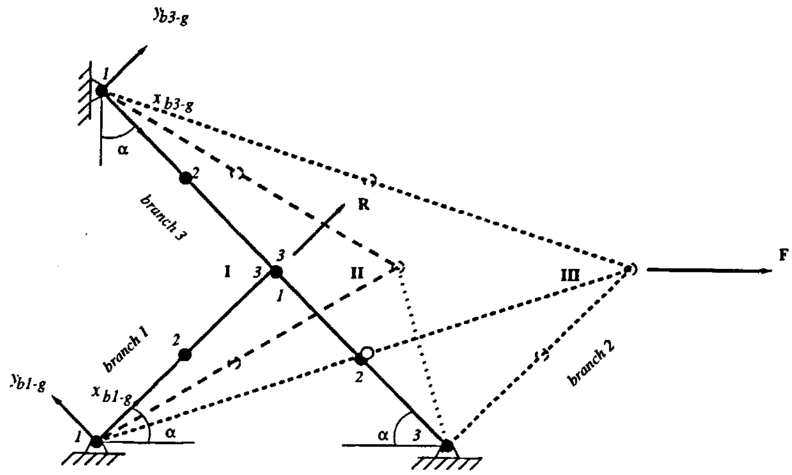


Figure 6.1: Cable construction illustrating negative stress in one cable during the deformation process

All three cables have length 100 $[m]$, stiffness parameter ($ae{=}AE$) 100.0 $[N]$ and **prestress** 5 $[N]$ to make sure that the initial state does not contain any tensionless cables resulting in singularities at the start of the computation. See appendix F for the used inputfile. As all three angles α are equal to 45^o the reaction forces of cable branches 2 and 3 compensate each other so that the total reaction force is 5 $[N]$ in the (longitudinal) x-direction of *branch1-global*.

The initial equilibrium state can be verified by checking the values of the *VECTOR* **FVAR.i.0**, representing the (internal) reaction forces at the first load step (**pa**=0). The same material properties are chosen as for example problem 5.3.1. An incremental load **F** is superposed at the center node in the horizontal direction, such that branch 2 will become tensionless at some stage of the computation. As before, this load is incrementally increased to a final value of 995 $[N]$.

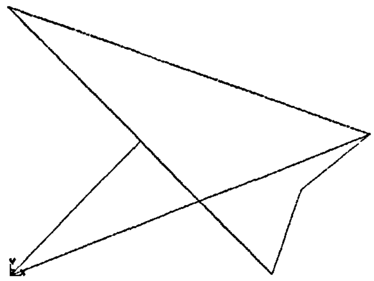


Figure 6.2: Deformed state at last computed load-step

A requirement of the method is to guarantee continuation of the computation during the stages with tensionless cable 2. However, as the cable elements connected to the interior node of cable 2 become tensionless, singularities occur at this node in accordance with the explanation in section 2.3. As a consequence, the stiffness matrix becomes singular and the computation stops prematurely at cycle 12, **pa** = 219.4975 $[N]$. See figure 6.2, representing the deformation of the cable structure at the last computed load step.

Notice the somewhat awkward position of the interior node of branch 2. As node 1 is joined with the center node it's displacement is prescribed. Also node 3 of branch 2 has a prescribed displacement (equal to zero), as it is fixed in all directions. The interior node however is not constrained and has no prescribed displacements. Hence, the displacement of this node is undefined when both cable elements connected to it become tensionless.

alternative: omit interior node
Obviously, one way to avoid this singularity problem eliminate the interior node in branch 2 and repeat the computation. Indeed, the computation is now not interfered by occurring singularities. The cable in branch 2 yields negative stress for load cycles 4 to 10, corresponding to **pa** values of 22.07107 $[N]$ to 191.0660 $[N]$.

The corresponding load-displacement diagram of the center-node (node 3 of branch 3) is presented in figure 6.3 and the corresponding numerical values of the final load-step are given in table 6.1.

VECTOR ***DISP.3.16***			
Node id.	**u**	**v**	**w**
1	0.0000E+00	0.0000E+00	0.0000E+00
2	1.4411E+02	1.2987E+02	0.0000E+00
3	2.8822E+02	2.5975E+02	0.0000E+00

Table 6.1: Results branch 3 (**pa**= 995.0 [N]) in branch3-global

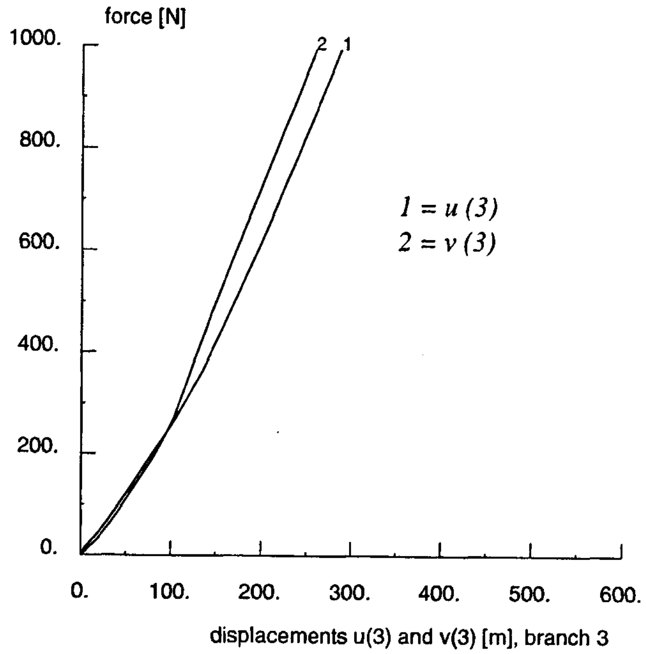


Figure 6.3: Load-displacement diagram of node 3, *branch3-global*

The displacements are given in *branch3-global*. A plot of the final deformation of the cable structure is given in figure 6.4. The shown disconnection of node 1 of branch 2 at the center-node is due to a deficiency in the **B2BASPL** post-processor. This is verified by comparing the load displacement diagrams of the center node for each branch. These coincide. Also notice the small difference between the curves 1 and 2 due to the presence of branch 2 and its contribution to the stiffness of the construction.

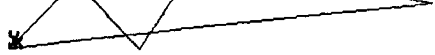


Figure 6.4: Deformation of cable structure without interior node in branch 2. Baspl view: disconnection

example problem 2: direct computation of cable sag; start problem
Another requirement the method must satisfy is to enable direct computation of e.g. a cable sag. Recall section 2.4, (part (v)). Obviously, the *AE-method* must also be tested for this property. For this purpose, consider the example problem of a cable submitted to transverse loading as discussed in section 5.3.4. The computation is now started directly from the initial unstressed state with cable length l_0, see figure 6.5 and appendix F for the used input-file.

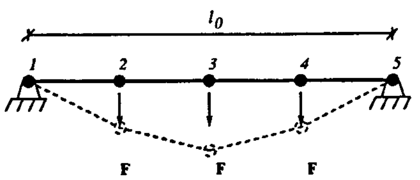


Figure 6.5: Simple cable example to illustrate cable sag computation by methods avoiding negative stress

Obviously, as the initial state is a tensionless state, the first and second variation are set equal to zero by the *AE-method* at initial load-cycle ($\lambda = 0$). This results in a singular stiffness matrix causing the computation to stop. Hence, the *AE-method* fails to solve the problem of an initially tensionless state.

6.1.4 Evaluation of results obtained by AE-method

The first example problem testing the method for a temporary tensionless (slack) cable as part of a construction shows that the *AE-method* enables continuation of the computation through these stages. However, the possibilities are limited, as singularities can still occur, causing the computation to stop or yielding un-realistic results.

Also nodes at which singularities can occur are hard to avoid if one wants to obtain somewhat accurate results from a finite element model. The *AE-method* thus solves the problem only partially. The last example which tests the start of the computation from initially unloaded states shows that the *AE-method* offers no solution in any way to this problem.

As an alternative the stiffness parameter can be assigned some small value for tensionless cables instead of being set equal to zero. This may solve the problem of singularities as discussed in example problem 1, but will obviously still fail to satisfy the requirement of direct computation from an initially unloaded state. Other methods need to be considered in order to find a satisfying approach to this problem.

6.2 Gravity or perturbation force method

6.2.1 Solution procedure for gravity-method

In addition to the *AE-method* another option is to make sure that all cable elements have tension stress at all times to avoid singularities. Contrary to previous sections stating that the effects of gravity can well be neglected, the presence of gravity can also be exploited to ensure that tension due to cable sag is always present in each cable element. The stretched cable possesses some resistance to transverse loads due to the induced transverse stiffness. Starting from the sagged configuration it is thus possible to incrementally apply an additional external load, without causing singularity problems.

In order to solve a problem by this method the computation needs to be split up in two parts. The first part computes the sagged configuration and corresponding reaction forces. These data are used to define the initial (deformed *and* loaded) state for the second part of the computation. Basically, the initial stresses are introduced in this second part with the definition of the new node coordinates from which the stretched cable lengths l_0^* ($\neq l_0$) are determined. The reaction forces required for equilibrium and the gravity forces are defined in **loadcase b** (constant values). Additionally, the external loads can be superposed incrementally by **loadcase a**.

The method described above does require some effort to define the model and create a second input-file. The issue raised by this method is whether it is possible to execute a computation by the above method and if it is, whether there are possibilities to compute the coordinates of the sagged equilibrium state implicitly by some implementations into the source code. This is desirable as obviously the creation of a second input-file is a time consuming and tedious work.

The next subsection will illustrate the procedure described in this subsection in more detail by some example problems.

6.2.2 Example problems to illustrate the gravity method

example problem, part 1: sagged equilibrium state

The first part is required to determine the initial (sagged) equilibrium state for the second part. The example used for this purpose is the symmetric triangular cable construction obtained by omitting branch 3 in example problem 1 of subsection 6.1.3. In this example the number of interior nodes is increased to obtain a geometric better approximation of the sagged configuration, see figure 6.6.

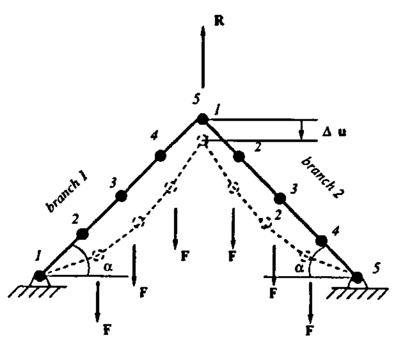


Figure 6.6: Symmetric triangular cable structure in sagged equilibrium state

In the first computation the initial elongation (prestress) is reduced to zero by use of prescribed displacements. The distributed forces **F** are applied simultaneously (see section 5.3.4). The data used in this example are given in table 6.2.

property	*value*
$\mathbf{F}_i$	$0.1\ N$
$l^*_{0_i}$	$25.0\ m$
$\Delta\mathbf{u}$	$5.0\ m$
l_0	$24.13230876\ m$
E	$100.0\ N/m^2$
A	$1.0\ m^2$

Table 6.2: Model data per element

The prestressed length l_0^* and corresponding displacement Δu are chosen for a given (total) initial length l_0 (= 96.52923505 [m]). Note that l_0^* and Δu are free to choose, as long as removal of $\Delta\mathbf{u}$ results in an initially unstrained configuration if gravity forces are absent. The initial length is easily obtained as follows:

$$l_0 = \sqrt{(l_0^* \cdot cos\alpha)^2 + (l_0^* \cdot sin\alpha - \Delta u)^2} \tag{6.1}$$

The reaction force **R** required for equilibrium of the prestressed state is obtained from:

$$\mathbf{R} = 2 \cdot EA \frac{l_0^* - l_0}{l_0} \sin\alpha \tag{6.2}$$

Substitution of the corresponding values yields:

$$\mathbf{R} = 5.084887352\ [N] \tag{6.3}$$

However, as use has already been made of prescribed displacements for the top-node, a reaction force need not and can not be prescribed on the same node. The input-file is presented in appendix F.

Before actually applying the gravity forces, a test run is executed to test whether the unsagged end position does indeed correspond to the unloaded state. This means that all reaction forces must approach zero as the computation approaches the final state. Indeed, the *VECTOR* **FVAR.i.n** approaches zero and the computation encounters singularity problems at the last loadstep.

Next, this computation is repeated *with* the gravity loads. The nodal positions are determined from the obtained results, see table 6.3. From these results a new equilibrium state can be defined in **loadcase b**. See part two for this

VECTOR **DISP.1.10**			
Node id.	u	v	w
1	0.0000E+00	0.0000E+00	0.0000E+00
2	-1.2311E+00	-4.9446E+00	0.0000E+00
3	-2.0138E+00	-6.8513E+00	0.0000E+00
4	-2.7129E+00	-6.2303E+00	0.0000E+00
5	-3.5355E+00	-3.5355E+00	0.0000E+00
VECTOR **FVAR.1.10**			
Node id.	u	v	w
1	0.0000E+00	0.0000E+00	0.0000E+00
2	-7.0711E-02	-7.0711E-02	0.0000E+00
3	-7.0711E-02	-7.0711E-02	0.0000E+00
4	-7.0711E-02	-7.0711E-02	0.0000E+00
5	8.9151E-01	8.9151E-01	0.0000E+00

Table 6.3: Results branch 1 (**pa**= 10.0) in branch1-global

second part of the analysis.

example problem, part 2: superposition of external load
From table 6.3 the new node coordinates are determined and defined in a new input-file (2). (See appendix F). The gravity loads **F** and corresponding reaction force **R** on the top-node are defined as constants in **loadcase b**. From the table one can see from the *VECTOR* **FVAR.1.10** that (the opposite of) the reaction forces on the interior nodes correspond to the imposed external loads **F** and that the reaction force at the top-node is equal to

$$\mathbf{R} = \sqrt{2 \cdot (0.89151)^2} = 1.260785533 \ [N]$$

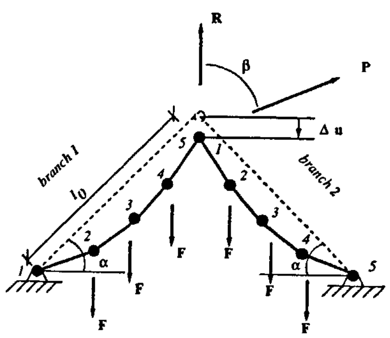
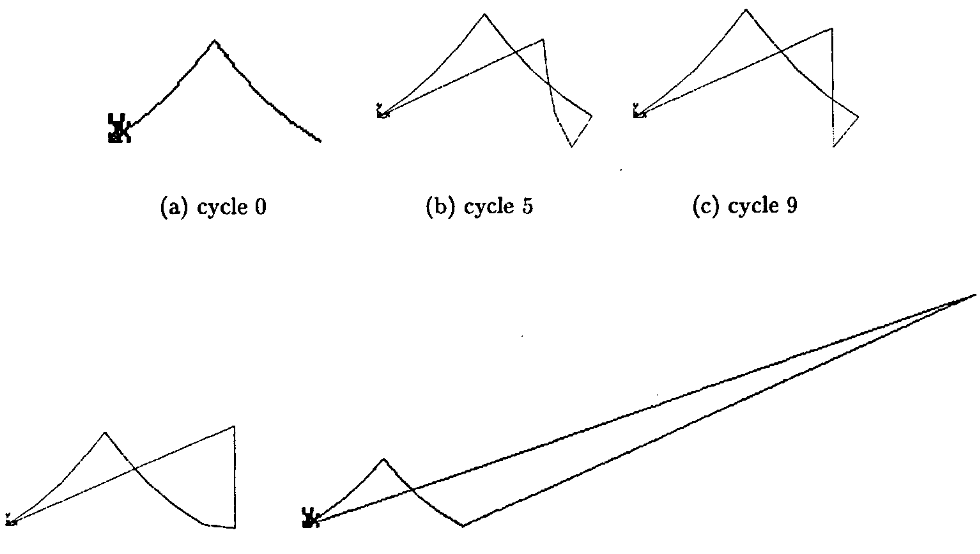


Figure 6.7: Sagged equilibrium position used as initial stressed state

Recall from chapter 3 that either load or displacement may be defined on one single node. Hence, with the sagged state defined as the initial equilibrium state, one can now superpose an external load **P** on the top-node in **loadcase a** of the new input-file. The direction of **P** is determined by the angle $\beta = 22.5^o$. See figure 6.7. The load **P** is increased incrementally to an end value of 995.0 $[N]$ by executing a restart after the conventional run with **B2CONT**.



(a) cycle 0 (b) cycle 5 (c) cycle 9

(d) cycle 22 (e) cycle 36

Figure 6.8: Deformation process of sagged cable structure submitted to force **P**

The deformation process is illustrated by figure 6.8 and the corresponding load displacement diagram of the top node in *branch2-global* is presented in figure 6.9.

As explained above, no negative stresses will be expected due to the presence of the constant loads **F**. Accordingly, the sagged structure deformes without singularities appearing at the nodes. However, it must be remarked that cable element 4 of branch 2 does get negative stress warnings for load-cycles 9 to 22, corresponding to **pa** 40.9 [N] to 84.65 [N].

6.2.3 Evaluation of results obtained by g-method

example problem part 1

For a simple cable construction the first part of the computation does not yield much problems. The definition of the model can be checked by omitting the external loads and reducing the initial elongations to zero to approach the unloaded state. The corresponding reaction forces are expected to approach zero (unloaded state).

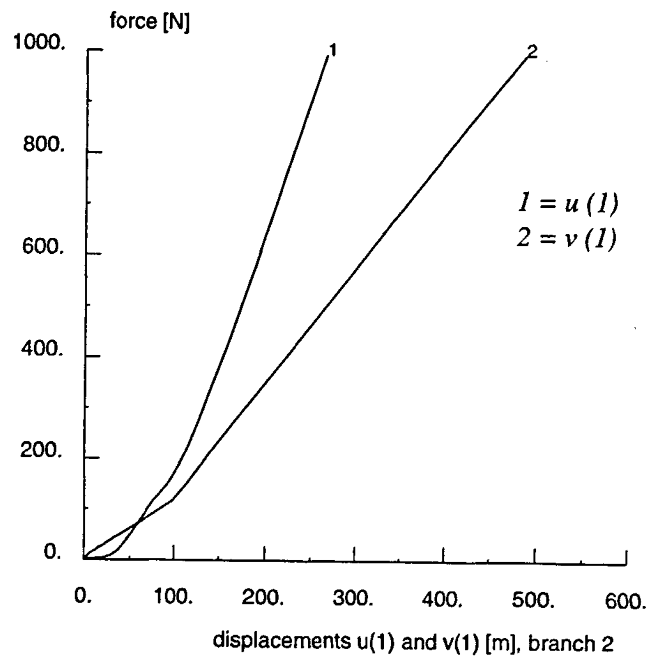


Figure 6.9: Load-displacement diagram of top node in branch2-global

example problem part 2
As expected, the second part of the computation does not yield any singularities. However, negative stress warnings are given for element 4 of branch 2 for a number of cycles. This was not predicted as the cables were expected to be in a state of tension throughout the computation due to the gravity forces. The lower cable element becomes tensionless due to a combination of the linear interpolation function used and the deformation or 'translation' of the upper elements.

Obviously, due to the linear interpolation the length of the lower cable element is directly determined by the distance between nodes 4 and 5. As node 5 is fixed to the earth and node 4 undergoes some displacement, this distance becomes smaller than the initial length l_0 of the element at certain stages of the deformation process.
Next, the translation part of the deformation process as shown in figure 6.8 can be explained as follows:

(A) Consider the situation as presented in figure 6.10. As long as node 3 is situated left of node 4, the load **F** can be carried by both cables, resulting in tension in both cable elements.

This can be explained by imagining what happens if element 4 is 'cut'. Obviously, node 4 will tend to move to the left, resulting in a pendulum mechanism. The presence of element 4 thus prevents such a mechanism to occur and is thereby submitted to tension. Consequently, the (tangential) stiffness matrix does not become singular.

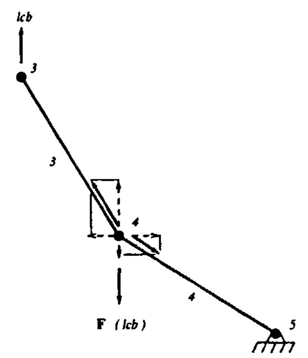


Figure 6.10: Situation A: both cables submitted to tension; equilibrium satisfied

(B) Consider the second part of the deformation process as presented in figure 6.11. From the instant that the position of node 3 is vertically above node 4, element 4 becomes tensionless and node 4 tends to move to the right. In fact a pendulum mechanism tends to appear As no equilibrium can be found for state II, the cable elements will translate one by one in the horizontal direction, see also figure 6.8, cycles 9 to 22. In accordance with part *(A)* element 4 has negative stress warnings for these cycles. However, as node 5 is fixed to the earth and the other element connected to node 4 is in tension, no singularities will occur.

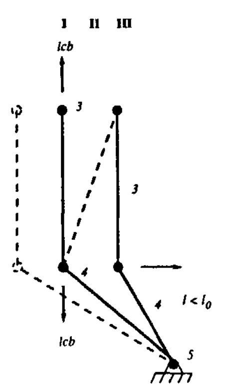


Figure 6.11: Situation B: the appearance of a pendulum mechanism

Apparantly, it is possible to compute the sagged equilibrium state prior to the computation with the actual external load, if the data for the unloaded state are given. However, the first part of the computation rapidly becomes more complicated as the number of cables increases and the definition of the input-file for the second part becomes an increasingly time consuming and tedious amount of work.

options to automize the transition from part 1 to part 2
Clearly, the most time consuming part of this method is the creation of a new input-file for the second part of the computation. It is thus desirable to investigate possibilities to automize this step, such that the new node coordinates and reaction forces are read in automatically and the gravity loads are automatically redefined in **loadcase b**. This allows the user to define one input-file only and to define the external loads directly in the automatically generated second input-file.

However, it must be stated that the first part of the computation does in fact not offer any solution for the 'start' problem as described in section 2.4, part *(v)*. Also, determining the value and direction of Δu for a more complex cable structure would still involve a large amount of work. Unless this is simplified or automized, it may be recommendable to first consider some other options, see sections 6.3 and 6.4.

6.3 l_0-method

6.3.1 Solution procedure for l_0-method

Another method that is worth investigating follows the same principle as the g-method, i.e. ensure that a small tensile strain remains as a lower bound for each cable element that becomes tensionless during the deformation process in such a way that the disturbance to the total stiffness of the construction is negligible. In other words, the tension must be small enough not to interfere with the behaviour of the structure as a whole, but large enough to guarantee continuation of the computation through the positions that physically exist.

As before, the element must follow the conventional computation at the instant it regains a state of tension. The method presented in this subsection satisfies these requirements. Contrary to the g-method, this method can be implemented directly into the element routine and thus does not involve complicated modeling procedures.

The procedure involves adjustment of the initial length l_0 for elements with zero or negative stress such that these elements are provided with a small tension. This occurs whenever the deformed length l becomes equal or smaller than the initial length l_0.

Please notice that this method ensures that the tangent to the solution curve at the singular freedoms corresponds to the direction of this tangent as soon as a state of tension arises again in the tensionless cables.

6.3.2 Modifications to element routine

Negative stress in cable elements manifests itself by a deformation resulting in a length l smaller than the initial length l_0. To provide a small tension proportional to the deformations, the initial length l_0 will be assigned a value slightly smaller than the deformed length l. This new initial length shall be referred to as $l_{0_{new}}$.

With the knowledge that the first and second variation are computed alternately in subsequent calls (recall subsection 5.1.2), it is obvious that prior to each first and second variation computation the initial length must be reset to it's original value.

Hence, Implementation of this method also requires a negative stress check prior to each first and second variation computation.

6.3.3 Property evaluation

This subsection will extend on the possibilities offered by this method. The adjustment of the initial length l_0 to $l_{0_{new}}$ allows direct computation with an initial unloaded state without the necessity of imposing a prestress. Obviously, it is desirable that both prescribed displacements and prescribed forces can be used to impose the external loads. The fact that this is indeed possible is explained in the sequel.

initializing the computation
Consider a cable construction with given geometry of the initial unloaded state submitted to external loading that will stretch the cables. See figure 6.12. The first cycle of the computation will obviously yield zero stress, as l_0 is initially equal to $(l =)\ l_0^*$. Consequently, the initial length l_0 will be redefined as

$$l_{0_{new}} = (1 - small) \cdot l \tag{6.4}$$

This yields an elongation of

$$\Delta l = l - l_0 = small \cdot l > 0.0 \tag{6.5}$$

resulting in a small positive strain ε of:

$$\varepsilon_{new} = \frac{small}{1 - small} \tag{6.6}$$

As soon as the cable stretches sufficiently in one of the subsequent load cycles (in this example the next load-cycle) the deformed length l will be found larger than the initial length l_0 and the computation continues the conventional route of positive stress for the remainder of the analysis using the original initial cable length l_0. Obviously, the results must coincide with the results obtained using prestress. This is illustrated in subsection 6.3.4 for the example of direct sag computation.

prescribed displacements
As a result of the first computation cycle (unloaded state), a small tension exists in the cables, see the explanation above. Consequently, a small reaction force (data set **FVAR.i.0**) is introduced to satisfy the equilibrium conditions. However, as the corresponding displacement w is equal to zero (!), a prescribed

displacement can be superposed incrementally from zero to the desired value without any problems. See also figure 6.12.

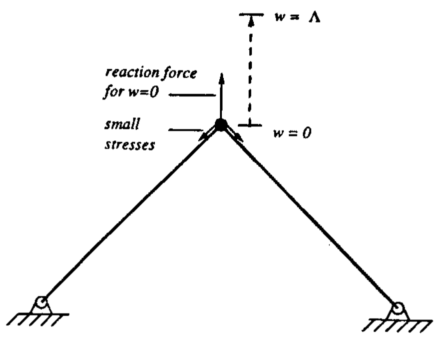


Figure 6.12: Prescribed displacement superposed on unloaded structure

prescribed forces
In case of prescribed force the problem arises of an initial reaction force at the top-node (**FVAR.i.0**), which is unequal to zero due to the small tension introduced (see above). For a correct computation the external load must be incremented from an initial value equal to this reaction force to the desired end value. The initial value is easily obtained from **FVAR.i.0** by executing a run prior to the actual computation. In most cases however, the initial reaction force will be small enough for the computation to converge to a state of equilibrium.

6.3.4 Example problems

example problem 1: cable as part of a construction
Consider the example problem defined in section 6.1. The analysis is now repeated with the element routine modified for the l_0-method.

Against expectations, the computation encounters some convergence problems and is forced to reduce the step-size to (very) small values causing the computation to stagnate at approximately **pa** $= 16.7$ $[N]$. By then the step-size is reduced so much that it is inpractical to continue because it becomes too time consuming.

The same problem occurs if the interior node in branch 2 is omitted. Again the computation stagnates at approximately **pa** $= 16.7$ $[N]$. Notice that this is sooner than the example in section 6.1.3, where negative stress occurs at **pa** $=$ 22.07017 $[N]$.

example problem 2: direct computation of cable sag: start problem
Consider the example problem described in section 5.3.4 of a cable submitted to transverse loading. The same data are used as mentioned in table 5.8. For direct computation of the cable sag the initial length l_0 is set equal to the length defined from the node coordinates; $l_0^* = l_0$. As expected, the computation is initialized and the results coincide with the results obtained in section 5.3.4, see table 6.4.

node id.	*x-disp*	*y-disp*
1	0.0000E+00 m	0.0000E+00 m
2	-1.1316E-01 m	-8.3590E-01 m
3	6.1736E-17 m	-1.1856E+00 m
4	1.1316E-01 m	-8.3590E-01 m
5	0.0000E+00 m	0.0000E+00 m

Table 6.4: Results direct cable sag computation

Comparing these results with the results obtained in section 5.3.4 by the displacement method and thus accounting for the prescribed horizontal displacements for the nodes 2, 3, 4, and 5 of respectively, 0.1, 0.2, 0.3 and 0.4 $[m]$ shows that the same deformed state has been found.

6.3.5 Evaluation of results obtained by l_0 method

Clearly, the $l_0 - method$ satisfies the requirements for direct computation of a deformed state with an initially unloaded configuration. Generally, the initial length will only require modification during the first load cycle to enable initialization of the computation.

However, for situations where negative stress occurs during a period of subsequent load cycles, the computation stagnates due to a strong reduction of the step-size. This can be explained by the following process:

The $l_0 - method$ introduces a small artificial positive strain. As a result we have:

(1) Small transverse stiffness

(2) A still large axial stiffness

Recall also the expressions for the second variation, eqs. (4.62) to (4.70). Point (1) is desirable. However, point (2) is completely wrong; the axial stiffness interferes severely with the structure, whereas the physical reality dictates that the tensionless cable becomes invisible for the rest of the structure. Hence, the $l_0 - method$ has actually changed the behaviour of the structure completely as soon as negative stresses occur with obviously adverse consequences.

The l_0-method solves the initialization problem but introduces a problem for continuation of the computation for multi-cable structures involving cables becoming tensionless during the deformation process (for a number of load cycles). Thus further investigation is required. Section 6.4 combines the benefits of the *AE-method* with the benefits of the l_0-*method.*

6.4 Combined l_0-E method

6.4.1 Solution procedure for combined l_0E method

Combining the benefits of both the l_0 and AE method, another approach to the negative stress problems can be extracted. The procedure of this method is set out below.

l_0 method
The method of l_0 described and tested in section 6.3 offers a solution to the initialization problem. However, continuation of the computation for a temporary tensionless cable as part of a construction turned out to be a problem. Recall example 1 of subsection 6.3.4. The idea behind the method is comparable with the method provided by the original element routine. In the original element routine, $\Delta l = l - l_0$ was set equal to zero for negative stress. Obviously, this does not offer possibilities to solve the start problem. By setting Δl equal to some small value instead, it is possible to start the computation from an unloaded state. However, in both cases the cable still has a considerable axial stiffness left, as can be seen from the expressions for the second variation, (4.62) to (4.70).

AE-method
The *AE-method* described and tested in section 6.1 on the contrary, does not solve the initialization problem (example 1), but it does offer some possibilities for continuation of the computation (example 2). As was described in this section, singularities may occur at the instant that the cable elements connected to a node become tensionless and ae is set to zero for these elements yielding a singular stiffness matrix. Similar to the l_0 method, the stiffness parameter ae can be assigned a small value instead. This way the stiffness matrix does not become singular.

Both methods described in the above are incomplete. The methods do however offer complementary solutions to the problem as a whole:

Multiplying the components of the element stiffness matrix by some small value by assigning ae some small value, in fact corresponds to providing the tensionless cable elements with a small axial (compression) stiffness. Consequently, the tensionless cables will behave like rubber elastics and mainly just adapt to the geometry of the remaining structural elements without significantly interfering with the deformation process.

When prescribed forces are used, the small stiffness may result in large displacements. As soon as negative stress occurs, the $l_0 - method$ ensures a small positive strain (as lower bound of possible strains). To eliminate the still large axial stiffness, a small value for AE is also introduced. Now the tensionless cable behaves like a very soft (rubber-like) structural element that no longer interferes with the structure. However, all degrees of freedom remain non-singular.

6.4.2 Modifications to element routine

As before, implementation of the method requires a negative stress check prior to each first *and* prior to each second variation computation. In case of negative stress the initial length is redefined and the stiffness parameter ae is multiplied by a small factor. To ensure that the correct values are read in each cycle, both l_0 and AE are reset to their original values prior to each first and second variation computation.

6.4.3 Example problems

example problem 1: cable as part of a construction (1)
The new element routine is tested on example problem 1 of section 6.1, see also figure 6.1. With interior nodes in each branch the computation continues without any problems and stops at the final value of **pa** = 995.0 [N], as prescribed. Negative stress warnings are given for load-cycles 4 to 10 corresponding to **pa** 22.07107 [N] to 191.0660 [N]. This corresponds to the same region of negative stress as was found in subsection 6.1.3. The corresponding load-displacement diagram is given in figure 6.13 and the numerical results are presented in table 6.5.

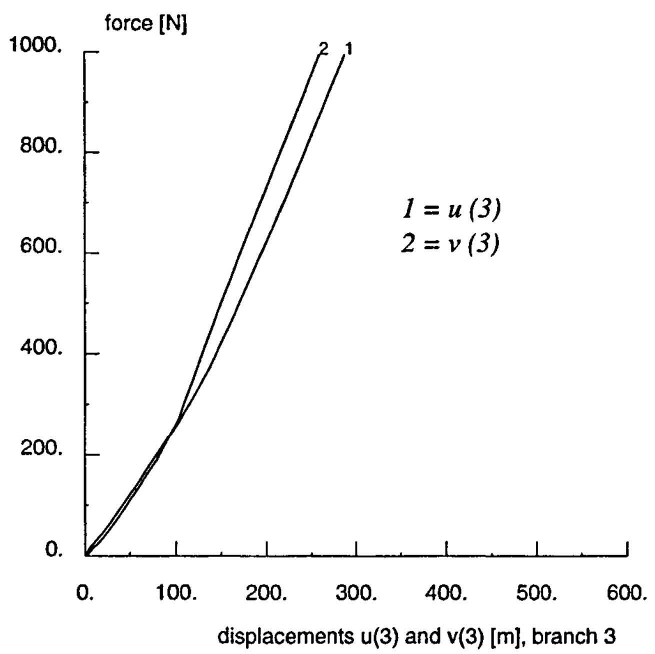


Figure 6.13: Load-displacement diagram of node 3 branch 3

Repeating the computation on the same model with omission of the interior node in branch 2 yields the exact same deformed state.

From the results in table 6.1 and the load-displacement diagram in figure 6.13 one can see that the results coincide and that the presence of the interior

VECTOR DISP.3.16			
Node id.	**u**	**v**	**w**
1	0.0000E+00	0.0000E+00	0.0000E+00
2	1.4411E+02	1.2987E+02	0.0000E+00
3	2.8822E+02	2.5975E+02	0.0000E+00

Table 6.5: Results branch 3 (**pa**= 995.0 [N]) in branch3-global

node in branch 2 is no longer a problem.

example problem 2: direct computation of cable sag: start problem
Finally, the modified element routine is required to enable direct computation of e.g. a cable sag. For this purpose example problem 3 of subsection 6.1.3 is used. As expected, no problems are now encountered at the start of the computation. The results presented in table 6.6 correspond to the cable sag obtained in subsections 5.3.4 (using prescribed displacements) and 6.3.4 (l_0-method).

node id.	*x-disp*	*y-disp*
1	0.0000E+00 m	0.0000E+00 m
2	-1.1316E-01 m	-8.3590E-01 m
3	1.6783E-16 m	-1.1856E+00 m
4	1.1316E-01 m	-8.3590E-01 m
5	0.0000E+00 m	0.0000E+00 m

Table 6.6: Results direct cable sag computation by l_0E method

6.4.4 Evaluation of results obtained by combined l_0E method

The combination of the l_0-method and the $ae = small$-method appears to satisfy the requirements set on the cable in section 2.4. By this method it is possible to compute a deformed state directly from an initial unloaded state. This may be very convenient, especially where structures consisting of several cables are to be analyzed.

Additionally, as tensionless cables will behave like rubber elastics the problem of singularities has been eliminated as well. As a consequence, the effect of these cables on the deformation process is negligible and the computation continues until the deformed state, corresponding to the final value of the imposed external load, has been obtained.

Although the start problem seems to be solved, it must be remarked that in the first iteration cycles the displacements become excessively large when prescribed forces are used. This is because in addition to the l_0 modification, the (axial) stiffness parameter ae is given a small value resulting in a small axial stiffness, i.e. small resistance in the axial direction. As a consequence, a large number of iterations is required for convergence of the first load-step, i.e, if convergence is achieved at all. A new strategy to combine the l_0*−method* and the AE*−method* such that this problem does not occur will be explained in section 6.5.

6.5 Refined l_0E method

6.5.1 Solution procedure for refined l_0E method

From the previous section it has become clear that the combined l_0E *− method* does not provide a complete solution to the problem. The small (total) stiffness at the start of the computation results in excessively large displacements such that a large number of iterations is required for convergence of the first step.

Also, taking into account the findings in sections 6.1 and 6.3 it becomes obvious that there are two situations that need to be regarded separately; the continuation problem and the start problem (recall section 2.4 parts *(iv)* and *(v)* respectively). This means that in order to solve both problems effectively both solution methods need to be implemented separately as well.

It must be realized that a combination of both problems is also possible: To illustrate this, consider again the (multi-)cable construction from section 6.1. (See figure 6.14).

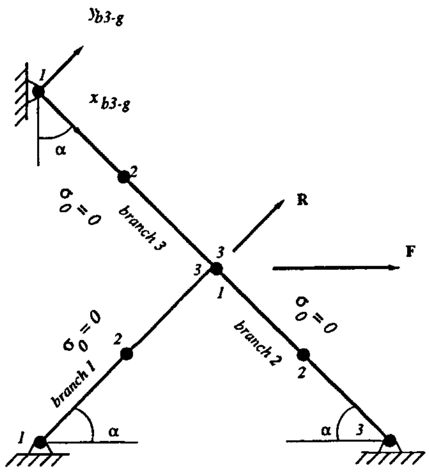


Figure 6.14: Cable construction used to illustrate combination of two solution methods

Assume all cables to be initially tensionless. By incrementally increasing the external force **F**, the cables in branches 1 and 3 will assume a state of tension, but the cable in branch 2 will (at first) remain tensionless. The start situation of the cables in branches 1 and 3 is similar to the example problem discussed in section 6.3 (example 2). The desirable solution method to this problem was shown to be the l_0 *− method.*

However, applying the $l_0 - method$ also for branch 2 would be incorrect; branch 2 would conserve a relatively large axial stiffness and thus interfere with the deformation of the other structural elements. If only the $AE(= small) - method$ would be applied, the presence of interior nodes in branch 2 would cause singularities (due to $\Delta = 0$ components, see eq.(4.71)), which is also not desirable. This problem is avoided by *superposition* of the $l_0 - method$, ensuring a small transverse stiffness and so avoiding singularities.

The example described above illustrates that it is desirable to enable specification of the solution method for each cable (branch) separately. To achieve this the following two options must be made available:

(A) The first load-step is executed by the l_0 method. The tensionless cable assumes a state of tension after this first load-step and the computation is continued with the original value of the initial length l_0. Obviously, this way large displacements at the first load-step due to a low axial stiffness are prevented. If the cable becomes tensionless again at some point of the computation the combined $l_0E-method$ is applied to ensure continuation of the computation.

(B) (Default method). This second option becomes the default method, which applies the combined $l_0E - method$ as described in section 6.4 for tensionless cables. This method is suitable for most other problems excluded the problems described in (A).

6.5.2 Modifications to element routine

The two methods (A) and (B) as described in the previous subsection, are both made available to the user by enabling the user to request the suitable method per cable element according to the user's own insight.

The default method can be requested by defining an initial length $0 <$ **l0** $< l_0^*$ or initial stress **prestress** ≥ 0 as usual. Specification of initial length **l0**$= 0.0$ is (obviously) not possible and will result in an error message.

By specifying an (arbitrary) *negative* value for either the initial length **l0** or the initial stress **prestress** the user requests the $l_0 - method$ for the first load-step (start), after which the computation will be continued by its default method. Proper use of both options will be explained in more detail in subsection 6.5.3.

The second component of the array of directives *ecckern(i)*, i.e. *ecckern(2)* has been activated for the purpose of recognizing the selected method. For values of initial length **l0** > 0.0 and for values of initial stress **prestress** ≥ 0.0, the directive *ecckern(2)* is given a positive value ($\geq +1.0$), indicating the default method. If a negative value is given for **l0** or **prestress**, *ecckern(2)* is given a negative value (≤ -1.0), requesting to start with the $l_0 - method$. If a rod element is specified, *ecckern(2)* is set equal to zero.

The array of pre-variational values, *elprev(i)* is extended by one component *elprev(6)* used for recognition of the $l_0 - start - method$ (*elprev(6)=-1.0*) in the element routine *b2ep39.F*. For the default method *elprev(6)* is assigned the value 1.0.

The modified (pre-variational) element routines *b2epv39.F* and *b2ep39.F* are presented in appendix E. Notice that after the first load-step all values are reset for the default method.

6.5.3 New user specified options

With the implementation of the separation of the two methods, it is left up to the user to specify the desired method per cable. The selected method depends on the values specified for **l0** and/or **prestress** according to table 6.7.

l0			*prestress*		
< 0.0	$=0.0$	> 0.0	< 0.0	$= 0.0$	> 0.0
l_0-start	Error	Default	l_0-start	Default	Default

Table 6.7: User specified method

The user must take into account some directions when using the available options.

- The $l_0 - method$ is suitable for computations with initially tensionless cables, which assume a state of tension after the first load-cycle.
- If a cable is initially tensionless and will (at first) remain tensionless during the deformation process the default method is most suitable. (Specify **prestress** = 0).

In both cases the user must take care to define the node coordinates consistent with the initial length l_0. This is because for these cases the value used for the initial length l_0 is equal to the cable lengths following from the given node coordinates (*elprev(4)*). Obviously, for most other cases the default method will be suitable.

6.5.4 Example problems

example problem 1: combined (multi-)cable construction
To illustrate the use of the implemented options, an example problem will be considered which combines the 'continuation' problem and the initialization problem. See figure 6.14. All cables are taken to be tensionless in the initial configuration. By inspection one can see that the cables in branches 1 and 3 will assume a state of tension after the first load-increment, and that the cable in branch 2 is initially tensionless, but will remain tensionless during the first load-cycles.

Consistent with the directions given in subsection 6.5.3 the $l_0 - start - method$ will be requested for the cables in branches 1 and 3 and for the cable in branch 2 the default method. The force **F** is increased incrementally to a value of 1000.0 $[N]$ in 16 load-steps. The corresponding load-displacement diagram of node 3 of branch 3 is presented in figure 6.15.

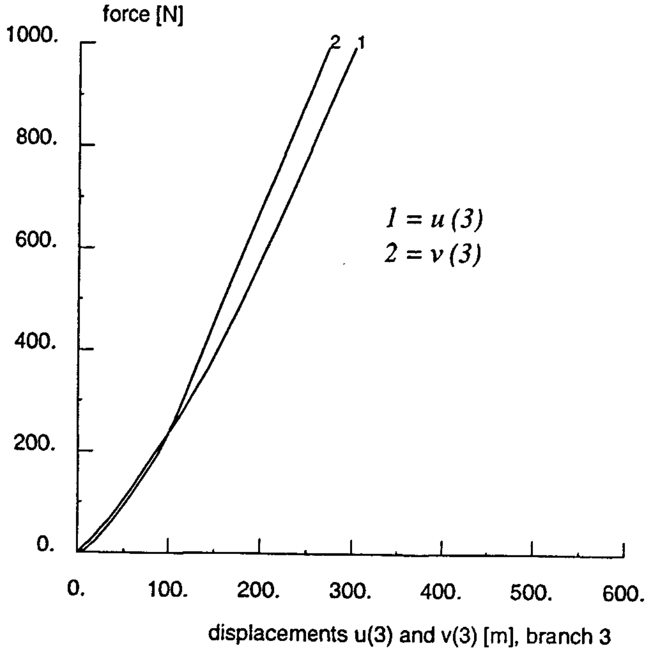


Figure 6.15: Load-displacement diagram of node 3, branch 3 in *branch3-global*

As can be seen clearly from the figure, the cable in branch 2 initially has negligible interference with the other cables; the displacement curves 1 and 2 practically coincide, indicating the symmetric problem which is obtained if branch 2 is omitted. The computation starts without any problems and continues problem-free through the cycles with tensionless cable in branch 2. As **F** continues to increase, branch 2 assumes a state of tension and contributes to the stiffness of the cable structure, see figure 6.15.

Also notice that the load-displacement diagram in figure 6.15 shows that node 3 of branch 3 ends with slightly larger displacements than the load-displacement diagram in example 1 of section 6.4. Obviously, this is because in section 6.4 the computation is started with prestressed cables.

example problem 2: direct computation of cable sag; start problem
For completeness the method is also tested on the example of direct cable sag computation. For this purpose a negative value is specified for the initial length **l0** or initial stress **prestress**. The obtained results coincide with the numerical results presented in section 6.3, table 6.4. This comes as no surprise as in both examples the first load-step uses the $l_0 - method$ after which the cable

is in tension and the computation is continued by the conventional procedure ($\sigma > 0$).

6.5.5 Evaluation of results obtained by refined l_0E method

The new approach to treat the two problems of continuation of the computation and the start problem separately yields satisfying results. Both options can be specified by the user per cable as desired. This way a large variety of problems involving cable structures can be solved. Combination of the two methods was illustrated in subsection 6.5.4. It was found that no large displacements will occur at the start of the computation and convergence requires only a small number of iterations. Also the default method (combined $l_0E - method$) does not result in any problems during the cycles with tensionless cable in branch 2.

The method presented in this section can thus be regarded as a workable approach to the singularity problems in quasi-static analysis. The approach is consistent with the actual mechanical behaviour of the cable and allows a broad variety of problems to be solved.

One important remark remains to be made:
For the cable subjected to compression alternative formulations are used for the first and second variations. Hence, it is important for the user to realize that in situations where all transverse degrees of freedom of the cable are locked, the cable is stabilized. (Recall sections 2.2 and 2.3). The element to be used in such cases will thus be a rod instead of a cable.

Chapter 7

(NLR) Mast example: a practical application

Construction problems may be defined in various ways. Depending on the given data certain problems may arise within the definition of the finite element model required for the desired structural analysis. This may be a continuation, dynamic, buckling (failure) or optimization analysis. It has been attempted to bring in cable properties useful to situations expected to be most commonly encountered with the definition of a finite element model. Recall section 2.4 and chapter 6.

This chapter treats the problems occurring with the definition of a mast model as an illustrative example of a practical application of the developed non-linear cable element. For this purpose a simple model is extracted from a composite mast of a sailing yacht [10].

7.1 Introduction (to the Mast problem)

The example used is a 57 m long composite mast of a sailing yacht. The design and fabrication technology was developed by the 'Structures Technology' department of the National Aerospace Laboratory (NLR) under contract of Royal Huisman Shipyard. The development of the design and fabrication technologies involve several aspects. In this chapter the attention will be focused on the problems arising with the definition of a mast model for executing a (buckling) failure analysis.

The development of the design and fabrication process of the mast in consideration consists of four building blocks:

a) A selection of fabrication processes and materials

b) Numerical design of the structure

c) Actual fabrication of the mast

d) Testing

Fabrication of a composite mast, as was designed by the *NLR*, instead of a conventional aluminium one, has lead to a total weight saving of approximately 800 *kg*.

Obviously, this chapter will focus on block b); 'Numerical design of the structure'. The numerical analysis required, comprises an optimization analysis and a buckling failure analysis. In order to perform these analyses a proper insight to the procedure of mast loading needs to be acquired. The loading aspects of the mast will be treated in section 7.2. For this purpose an illustrative simple mast model will be considered, see figure 7.1.

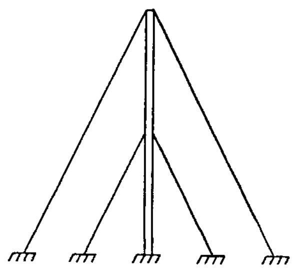


Figure 7.1: Simplified mast model

For the numerical optimization analysis the design constraints are the buckling loads and stresses. The design variables are the ply thicknesses of the composite mast skin. Again, attention shall not be focused on the optimization of the ply-thicknesses. Instead, some attention will be given to the implications involved with the optimization of the initial cable lengths and/or pre-stresses required to obtain the desired '*harbour condition*'. This condition is described in section 7.2.

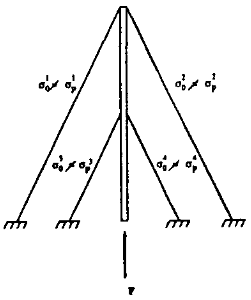
7.2 Loading conditions of the mast

The total loading on the mast consists of two parts: the *harbour condition* and the *sailing loads.*

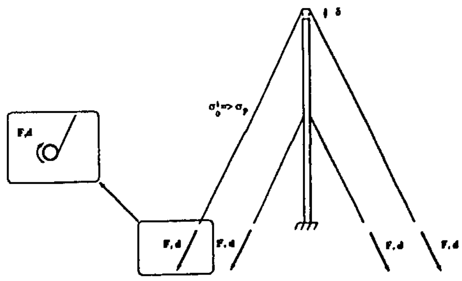
harbour condition

In the '*harbour condition*' the mast is in a state of pre-compression and the cables are in a state of pre-tension. The pre-tension in the cables must be sufficiently large to avoid compressive loads in the cables at any time during *sailing conditions.*

In practice the mast is brought into *harbour condition* as follows:
First the rigging is attached to the mast. The mast is then jacked upwards by a hydraulic mechanism. As a consequence, tension loads are introduced into the cables. However, the tension loads in the cables obtained by this procedure generally do not correspond with the tension loads required for the desired *harbour condition.* Therefore the cables are 'tuned' by changing the cable lengths. The



(a) jacking of the mast introducing tension in cables



(b) tuning of cables by changing cable lengths

Figure 7.2: Establishing the harbour condition by respectively jacking up the mast a) and tuning the cables b)

final shape that is obtained by this method is known as the *'harbour condition'*. The two stages described above to obtain the *harbour condition* are illustrated in figure 7.2.

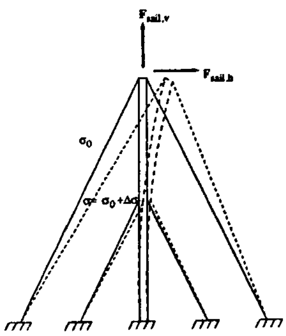


Figure 7.3: Superposition of sailing loads

sailing loads
Obviously, the *sailing loads* are the loads to which the mast is submitted during sailing. The determined worst case resulted in a compressive load of approximately 200 *tons* at the foot of the mast. The (total) deformation of the mast due to the *sailing loads* is obtained by considering the *harbour condition* as initially loaded *and* deformed state and superposing the *sailing loads* onto this *harbour condition*, see figure 7.3.

7.3 Problem definition

The main problems of the numerical analysis that were encountered by the *NLR* were:

(i) The cables were modeled by linear beam elements. Obviously, a better numerical approximation to the problem is achieved by using (geometrically) non-linear cable elements instead. Additionally, the present formulation of the cable element will save computation time. With the introduction of the developed non-linear cable element the possibility of pre-loading in the form of pre-stress has been introduced. See also chapter 2.2. This offers a range of new possibilities to approach the problem.

(ii) Another problem encountered with the definition of the finite element model was the definition of the *harbour condition.* As the initial unloaded geometry of the mast-structure is not known, the loaded *and* deformed *harbour condition* must in fact be used as initial condition for the analysis. However, for the definition of the finite element model the problem arised of the **B2000** code failing to provide the option of prescribing a pre-load on the used linear beam elements.

Instead, a deformed *un*loaded condition was used as initial state for the analysis. However, as stated in *(i)* the use of cable elements does allow for the definition of a pre-loading (pre-tension).

(iii) The tuning of the cables after the jacking of the mast introduces a modeling problem as well. As the cables are tuned afterwards the cable lengths are changed to obtain the desired configuration. However, within the optimization procedure it is not feasible to include this change of cable lengths into the analysis, see also ref. [10].

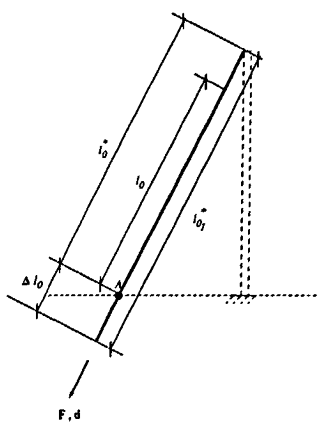


Figure 7.4: Interpretation of mast tuning

The change in cable lengths due to the tuning of the mast can be interpreted as illustrated in figure 7.4. The initial (unknown) unloaded cable length is represented by l_0 as usual. The length of the cable after jacking up the mast is l_0^* and the 'virtual cable length' due to the tuning of the cable (see also figure 7.2) is given by $l_{0_1}^* = l_0^* + \Delta l_0$. The elongation Δl_0 is brought into the cable by using prescribed displacements or corresponding prescribed force. The final (pre-)stress in the cable can be represented by :

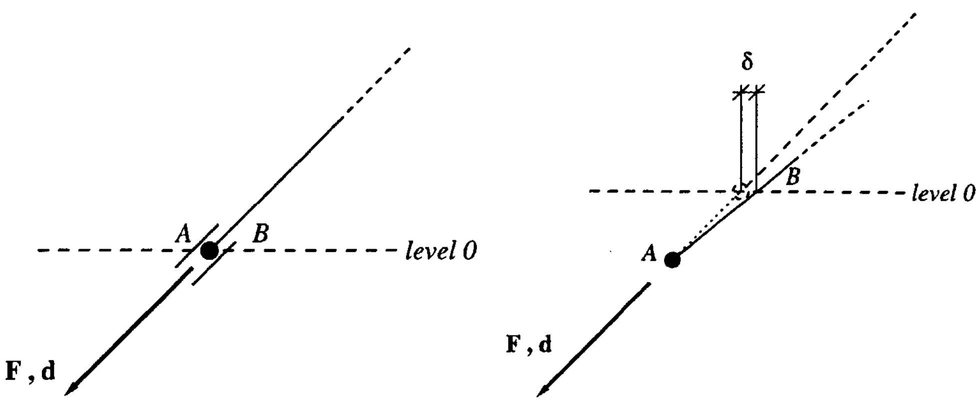
$$\sigma_{0_1} = \frac{l_{0_1}^* - l_0}{l_0} = \frac{l_{0_1}^* - l_0^*}{l_0} + \frac{l_0^* - l_0}{l_0} \tag{7.1}$$

or

$$\sigma_{0_1} = \Delta\sigma_0 + \sigma_0 \tag{7.2}$$

Clearly, the change of pre-stress is dependent on Δl_0.

One of the problems arising is the definition of the boundary conditions on node A. As the deformed length of a cable element is conventionally computed from the node positions it may become problematic to compute the deformed length $l_{0_1}^*$ in combination with the boundary conditions at node A. A sliding support could be introduced as boundary condition at node A. However, as the mast and cables deform, the boundary condition no longer holds at the connection level (represented by 'level 0'), see also figure 7.5.



(a) locking transverse degree of freedom at node A

(b) discrepancy at connection cable-ship due to cable/mast deformation

Figure 7.5: Boundary condition on connection node A

As can be seen from the figure, the connection point B and the element node A do not coincide after deformation by a discrepancy δ.

(iv) Also the *sailing loads* must be superposed properly for the (buckling) failure analysis, i.e. superposition onto the *harbour condition.* This also involves some modeling implications as will be discussed in section 7.4.

(v) Finally, the large number of cables connected to the mast must be taken into account.

The transverse loads are introduced into the mast at the spreaders, the sail attachment points and the pulleys. The analysis as has currently been done by the National Aerospace Laboratory (NLR) was performed on a simplified analysis model. The following assumptions were made to obtain this model:

- The difference in deformation due to the application of the *harbour condition* as an unloaded geometry in the finite element model instead of a loaded geometry was assumed to be tolerable.
- The rigging keeps the mast in its place at the rigging attachment points, allowing the rigging to be replaced by sliding supports at these locations. This way a simplified model was constructed for the buckling analysis instead of dealing with the problem of changing cable lengths while tuning the rigging.

Taking into account the stated model definition problems an attempt shall be made to propose a solution strategy which will make the above assumptions abundant.

7.4 Proposed problem approach

First of all, it can be stated that replacing the beam elements by cable elements introduces the option of pre-loading in the form of prestress. In order to investigate the possibilities offered by this option with respect to the mast definition, a proper interpretation of the prestress option is required.

prestress
The prestress option as has been implemented into the code can be interpreted as follows:

It is not sufficient to simply define the prestress(es) belonging to the desired deformation of the mast (*harbour condition*). One must work towards this deformed position of the mast from an initially undeformed, but prestressed state. The reaction forces required to equilibrate the prestressed condition should be delivered by the mast.

problem approach
In order to obtain the deformed structure using this pre-stress definition some 'tricks' must be applied.
Consider the simplified mast-structure presented in figure 7.6. The proposed method will be illustrated by the tuning procedure of one (single) cable.

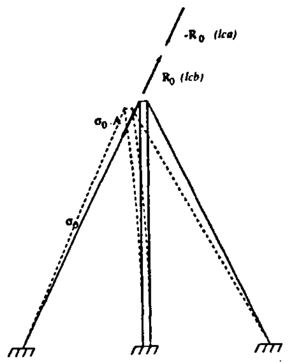


Figure 7.6: Proposed method to obtain deformed *and* loaded harbour condition

With the prestress option being available, a prestress can be defined on the cable in question by specifying either **prestress** or **10** in the input-file. The reaction force required to equilibrate the pre-stress, $\mathbf{R}_0$, will be defined in **loadcase b** and is held constant. As stated above, this will not result in a deformation of the structure. By defining an incremental force opposite to this reaction force in **loadcase a**, the reaction force in **loadcase b** can be reduced to zero incrementally. As $\mathbf{R}_0$ is reduced, reaction forces appear in the mast at the attachment points, taking over the task of equilibrating the pre-stress in the cable. See also figure 7.7.

The result is a deformed and loaded configuration which represents the *harbour condition* and can be used as initial state for the finite element (buckling) analysis. The method does however introduce a few other problems. These problems and some suggestions to approach these problems will be discussed in the next section.

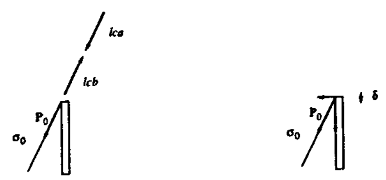


Figure 7.7: Reaction forces on mast

7.5 Validation of the proposed method

Due to the change in **loadcase a** the whole structure takes on a deformed state. One can vary these values until the desired *harbour condition* is obtained. However, two problems may arise using this method.

1) The pre-stress σ_0^* finally obtained in the cable, (i.e. after tuning) will possibly not coincide with the prescribed stress σ_0, (equilibrated by $\mathbf{R}_0 =$

$\sigma_0 \cdot A$) due to the deformation of the mast. From the results it can be easily checked whether this discrepancy can be considered negligible.

2) Another consequence of the mast deformation, is that the cable changes direction and does not remain co-linear with the defined reaction force $\mathbf{R}_0$. Hence, the approach is only partially true if $\mathbf{R}_0$ is not completely compensated by **loadcase a**. See also figure 7.8. Apparantly we are dealing with a *live load* problem. Hence, in order to perform this procedure accurately one needs to define the loads as 'live loads'. Unfortunately, this option is currently not available in **B2CONT**.

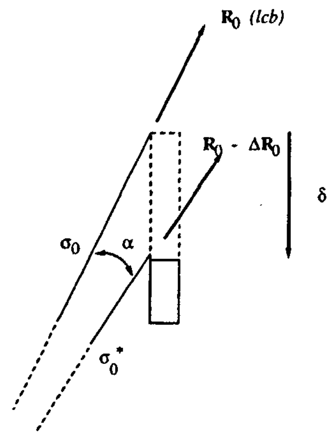


Figure 7.8: Discrepancy due to mast deformation

The discrepancies described in part *(2)* can be expected to be negligible. However, it is possible to avoid these discrepancies by the method described below, based on the fact that the directions of the forces are no longer significant once the total equilibrating reaction force $\mathbf{R}_0$ on the node has been reduced to zero. The pre-stress in the cable and the corresponding reaction force are specified in the input file.

One can decrease the reaction force by use of **loadcase a** as described in the previous section, until the desired cable tension and/or deformation are obtained. The corresponding value of **pa** (**loadcase a**) is then used to redefine the problem, such that **lca** and **lcb** compensate each other, i.e. $\mathbf{pa} + \mathbf{pb} = 0$. This means that both loadcases can be removed from the computational database, without affecting the equilibrium state. For incremental superposition of the *sailing loads* one can now prescribe the *sailing loads* in the *VECTOR* **FRCA.GLOB (loadcase a**. This *VECTOR* can be obtained from a 'blanc analysis run'.

Finally, by performing a **B2CONT** analysis for this model the deformations due to *sailing loads* are obtained. By varying the values of the pre-stress in the cable, one can generate a $\sigma_0 - \mathbf{F}_{mast}$ dependency.

multi-cable construction problem
The method described above can in principle be used to perform the optimization and buckling failure analysis described in ref. [10]. However, the method requires some interactive steps, which complicates the analysis and is time consuming. This is specifically the case if a large number of cables are involved. If one or few cables have been tuned by the procedure, again one or few cables need to be tuned. This is still a complicated problem for optimization analysis. One cannot simultaneously reduce several reaction loads to zero by defining incremental forces in **loadcase a**, due to the non-linear load-displacement distribution along the mast.

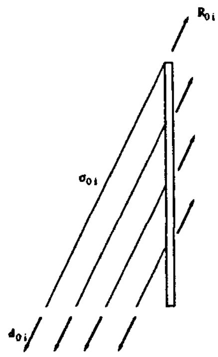


Figure 7.9: Tuning with prescribed displacements

Another option is to tune the cables using prescribed displacements assuming that the required displacements are small, see figure 7.9. The *sailing loads* can be superposed in the same manner as described before.

7.6 Recommendations

To facilitate the model definition and the computation for the method described in section 7.4 the following recommendations can be made:

1) Automize the required steps, such that the user only needs to define the desired pre-tensions and *sailing loads.*

3) Check whether the discrepancies described in section 7.5 can be considered negligible.

Chapter 8

Linearized vibrations around stable states

The second part of the assignment is to activate the code for linearized vibrations of cables around stable equilibrium states. Such analyses are performed by use of linear solution procedures, provided by the macro processor **B2LIN** in the **B2000** master code.

Linearized dynamic analysis requires a mass matrix for the element and its linearized stiffness matrix, i.e. the tangential stiffness matrix computed in the origin. An evaluation of the present formulation of the stiffness matrix will be given in section 8.2. The required modifications to the code are discussed in section 8.3. As an illustration to linearized vibration analysis around stable states, a numerical example problem is presented in section 8.4.

8.1 Linear dynamic analysis

8.1.1 Linearized vibrations around stable states

The (nonlinear) equations of motion in their discretized form can be derived by *Newton's* second law and are expressed by:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{f}(\mathbf{U}; t) = \mathbf{0} \tag{8.1}$$

with

$$\dot{()} = \frac{\partial}{\partial t}, \qquad \ddot{()} = \frac{\partial^2}{\partial t^2}$$

In the above equations (8.1) $\mathbf{M}$ and $\mathbf{C}$ represent the discretized mass and damping matrices respectively. The total force vector $\mathbf{f}(\mathbf{U}; t)$ is a function of the (discretized) displacements $\mathbf{U}$ and the time t.

Notice that for the static case ($\ddot{\mathbf{U}} = \dot{\mathbf{U}} = \mathbf{0}$) and ($\mathbf{U} = \mathbf{U}_0$), f becomes a function of the load parameter λ instead of time. Hence,

$$\mathrm{f} = \mathbf{f}(\mathbf{U}_0(\lambda), \lambda) = \mathbf{0} \tag{8.2}$$

Next, consider vibrations around an arbitrary stable equilibrium state determined by the displacements $\mathbf{U}_0(\lambda)$. This situation can be illustrated in a load-displacement diagram as presented in figure 8.1.

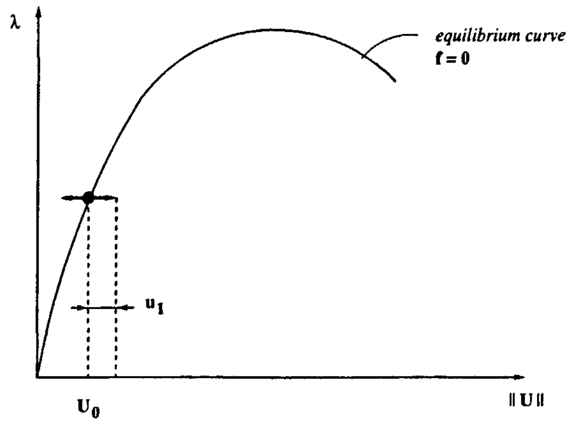


Figure 8.1: Linearized vibration around stable state

The (total) displacements $\mathbf{U}$ can be written as:

$$\mathbf{U} = \mathbf{U}_0 + \mathbf{u}_1 \tag{8.3}$$

where $\mathbf{u}_1$ represents small changes of displacements with respect to the stable equilibrium state under consideration. Substitution into eqs. (8.1) and taking into account that $\dot{\mathbf{U}}_0 = \mathbf{0}$ and $\ddot{\mathbf{U}}_0 = \mathbf{0}$, as it belongs to a stable equilibrium state, yields:

$$\mathbf{M}\ddot{\mathbf{u}}_1 + \mathbf{C}\dot{\mathbf{u}}_1 + \mathbf{f}(\mathbf{U}_0 + \mathbf{u}_1, \lambda + \Delta\lambda) = \mathbf{0} \tag{8.4}$$

Linearization of the above equations yields:

$$\mathbf{M}\ddot{\mathbf{u}}_1 + \mathbf{C}\dot{\mathbf{u}}_1 + \mathbf{f}(\mathbf{U}_0(\lambda), \lambda) + \mathbf{f}_{,\mathbf{u}_1}(\mathbf{U}_0(\lambda), \lambda)\mathbf{u}_1 + \mathbf{f}_{,\lambda}\Delta\lambda(t) = \mathbf{0} \tag{8.5}$$

with

$$\begin{aligned} ()_{,\mathbf{u}_1} &= \frac{\partial}{\partial \mathbf{u}_1} \\ ()_{,\lambda} &= \frac{\partial}{\partial \lambda} \end{aligned} \tag{8.6}$$

From eq.(8.2) one can see that $\mathbf{f}(\mathbf{U}_0(\lambda), \lambda) = \mathbf{0}$. Rewriting $\mathbf{f}_{,\mathbf{u}_1}(\mathbf{U}_0(\lambda), \lambda)$ as $\mathbf{K}(\mathbf{U}_0(\lambda), \lambda)$, the eqs. 8.5 become:

$$\mathbf{M}\ddot{\mathbf{u}}_1 + \mathbf{C}\dot{\mathbf{u}}_1 + \mathbf{K}(\mathbf{U}_0(\lambda), \lambda)\mathbf{u}_1 + \mathbf{f}_{,\lambda}\Delta\lambda(t) = \mathbf{0} \tag{8.7}$$

Clearly, the (linear) stiffness matrix $\mathbf{K} = \mathbf{f}_{,\mathbf{u}_1}$ is a function of the displacements $\mathbf{U}_0$, representing the (initial) stable equilibrium state considered and the load parameter λ;

$$\mathbf{K} = \mathbf{K}(\mathbf{U}_0(\lambda), \lambda)$$

The consequences of the dependency of the stiffness matrix on the initial state will be illustrated by considering the effects of prestress on the stiffness matrix, see the subsequent (sub)section. The eqs. (8.7) are solved in the usual manner using linear solution procedures.

Notice that for undamped free vibration analysis, the terms $\mathbf{C}\dot{\mathbf{u}}_1$ and $\mathbf{f}_{,\lambda}\Delta\lambda(t)$ in eq.(8.7) cancel, yielding:

$$\mathbf{M}\ddot{\mathbf{u}}_1 + \mathbf{K}\mathbf{u}_1 = \mathbf{0} \tag{8.8}$$

Conventionally, these expressions can be solved for the Eigen frequencies by assuming periodic functions for the displacements $\mathbf{u}_1$ as follows:

$$\mathbf{u}_1 = \hat{\mathbf{u}}_1 e^{i\omega t} \tag{8.9}$$

Substitution into eqs. (8.8) with the knowledge that $e^{i\omega t} \neq 0$ yields:

$$\left(\mathbf{K} - \omega^2\mathbf{M}\right)\hat{\mathbf{u}}_1 = \mathbf{0} \tag{8.10}$$

This set of equations can be solved by standard Eigenvalue analysis for the Eigen frequencies ω from which the corresponding Eigenmodes for the displacements, $\hat{\mathbf{u}}_1$, can be determined.

8.1.2 B2LIN

Linear static and dynamic (Eigenmode) analysis can be performed by the **B2000** macro-processor **B2LIN** by calling a sequence of processors. See for a more detailed description ref. [9].

The linear solution technique requires the linearized stiffness matrix and for dynamic analysis also the element mass matrix. The mass parameters can be specified in the input-file by using the **mass** command. With the parameters one can specify the elements, nodes and mass type [9] of the element. The mass type (**mt**) parameter specifies the mass generation to be selected; **LD** specifies lumped diagonal mass, **CD** consistent diagonal mass and **CO** generates consistent mass matrix.

The stiffness matrix will be treated in section 8.1.3.

8.1.3 Linearized stiffness matrix

Consider the nonlinear equilibrium path as was shown in chapter 3, figure 3.3. The stiffness matrix used for linear analysis is the linearization of the actual (tangential) stiffness matrix in the origin. This means that the stiffness matrix is a function of the initial geometry only. Hence, if **loadcase b** is used for the initial equilibrium state, **pa**=0 (zero displacements) in the origin. See figure 8.2.

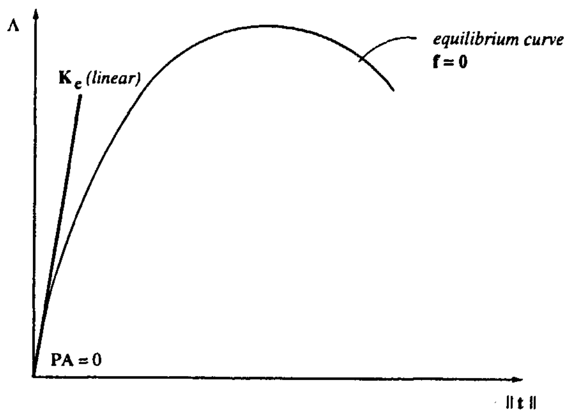


Figure 8.2: Linearized stiffness matrix used to compute the tangent at the origin

In linear analysis the linearized path that follows from linearization in the origin represents the equilibrium path considered in **B2LIN**. Obviously, as the displacements grow larger, the deviation between the linearized equilibrium path and the actual (nonlinear) equilibrium path increases.

Hence, linear analysis is only justified within a given region of small displacements. The corresponding stiffness matrix can be obtained directly from the nonlinear stiffnes matrix derived in section 4.6 by taking the values in the origin, i.e. setting the displacements equal to zero. This yields the following expressions for the linearized stiffness matrix:

$$\frac{\partial F_i}{\partial U_j} = \frac{\partial F_{i+3}}{\partial U_{j+3}} = -\frac{\partial F_{i+3}}{\partial U_j} = \frac{EA}{l_0}\left(\left(\frac{l_i}{l_0^*}\right)^2 (1-\Delta) + \Delta\right) \qquad i = j = 1,2,3 \tag{8.11}$$

$$\frac{\partial F_i}{\partial U_j} = \frac{\partial F_{i+3}}{\partial U_{j+3}} = -\frac{\partial F_{i+3}}{\partial U_j} = \frac{EA}{l_0}\left(\left(\frac{l_i}{l_0^*}\right)\left(\frac{l_j}{l_0^*}\right)(1-\Delta)\right) \qquad i,j = 1,2,3 \quad i \neq j \tag{8.12}$$

with

$$\begin{aligned} l_1 &= l_x \\ l_2 &= l_y \\ l_3 &= l_z \end{aligned} \tag{8.13}$$

and

$$\Delta = \frac{l_0^* - l_0}{l_0^*}$$
$$l_0^* = \sqrt{(l_x)^2 + (l_y)^2 + (l_z)^2} \tag{8.14}$$

See also section 4.4, eqs. (4.30) and section 4.5 for a further explanation of these expressions. Clearly, from the above expressions the presence of prestress results in an additional component $\Delta \neq 0$.

8.2 Validation of existing formulation

For linear analysis the directive **dirkern(3)**=1 and the other directives are set to zero. The current element routine then calls the following expressions for the stiffness matrix:

$$\frac{\partial F_i}{\partial U_j} = \frac{\partial F_{i+3}}{\partial U_{j+3}} = -\frac{\partial F_{i+3}}{\partial U_j} = \frac{EA}{l_0}\left(\frac{l_i^*}{l}\right)^2 \qquad i = j = 1,2,3 \tag{8.15}$$

$$\frac{\partial F_i}{\partial U_j} = \frac{\partial F_{i+3}}{\partial U_{j+3}} = -\frac{\partial F_{i+3}}{\partial U_j} = \frac{EA}{l_0}\left(\frac{l_i^*}{l}\right)\left(\frac{l_j^*}{l}\right) \qquad i,j = 1,2,3 \quad i \neq j \tag{8.16}$$

with

$$\begin{aligned} l_1^* &= l_x^* \\ l_2^* &= l_y^* \\ l_3^* &= l_z^* \end{aligned} \tag{8.17}$$

Note that for zero displacements ($\mathbf{U} = \mathbf{0}$) $l_i^* = l_i$. Obviously, as the stiffness matrix is symmetric, only the lower triangle components are required for the computation. In the existing element routine these equations ((8.15), (8.16)) are obtained from eqs. (4.62) to (4.70) of section 4.6 by setting $\Delta = 0$. However, by doing so, no additional terms appear that comprise the effect of a given prestress.

Consequently, no transverse stiffness is induced and an Eigenvalue analysis will only result in longitudinal modes instead of transversal modes. It requires no further explanation to see that a prestressed cable should yield transverse Eigenmodes due to the induced transverse stiffness.

Furthermore in the current formulation no negative stress or cable-rod checks are performed for linear analysis. Also no mass matrix is available in the current **B2000** code.

8.3 Modifications to existing formulation

modifications to stiffness matrix
Obviously, the existing formulation does not enable Eigenmode analysis of a pre-stressed cable. The linear stiffness matrix in *b2ep39.F* is modified for its correct formulation as presented in subsection 8.1.3. For linear analysis the second variation is computed only once at the beginning of the analysis, i.e. at the origin. Hence, the displacements are zero and thus the same expressions derived in section 4.6, eqs.(4.62) to (4.70) can be used for linear analysis. See appendix E for the modified element routine.

mass matrix definition
The element mass matrix is defined in the element mass routine *b2mp39.F*, see appendix I. This routine generates mass lumped diagonal, consistent diagonal mass or consistent mass according to the specified parameters. For the lumped mass generation the total mass of an element is computed and equally distributed over the two nodes. Note that the mass definition for the 2-node cable element can be determined directly from the rod-element mass matrix routine (*b2mp35.F*).

8.4 Numerical example problem

Eigenmode analysis
Finally, a numerical example will be presented to illustrate the Eigen frequency analysis of a cable. For this purpose use is made of the cable example in ref. [4], see figure 8.3.

This example involves a cable of (stretched) length 20 $[m]$, prestress $\sigma_0 =$ 500 $[N]$, density $\rho_0 = 0.3$ $[kg/m]$, and stiffness parameter $EA_0 = 2.2 \cdot 10^6$ $[N]$, discretized into 20 cable elements. To enable definition of the input-file the area A is chosen to be $2.2 \cdot 10^{-5}$ $[m^2]$, yielding for the density $\rho_0 = 1.3636$ $[kg/m^3]$ and for Young's modulus $E = 1.0 * 10^11$ $[N/m^2]$. See appendix G for the used input-file.

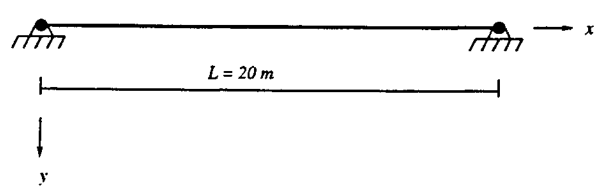
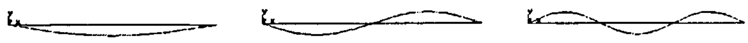


Figure 8.3: Stretched cable supported by two simple supports

The obtained converged Eigen frequencies are presented in table 8.1. The corresponding first three Eigenmodes are given in figure 8.4.

Analytical verification
The Eigen frequencies of a cable under prestress can also easily be determined analytically by the following expressions [1]:



(a) mode 1 (b) mode 2 (c) mode 3

Figure 8.4: First three Eigenmodes for cable with length 20 [m] and prestress 500 [N]

Mode	*Frequency*	*Omega*	*Eigval*	*Rel. Error*	*Status*
1	1.01957	6.40615	41.0388	1.731394E-15	CONVERGED
2	2.03285	12.7728	163.145	2.929669E-10	CONVERGED
3	3.03361	19.0607	363.311	2.992535E-08	CONVERGED
4	4.01565	25.2311	636.608	6.035404E-06	CONVERGED
5	4.97295	31.2460	976.311	6.133633E-05	CONVERGED

Table 8.1: Results Eigenmode analysis: converged Eigenmodes

$$\begin{aligned} f_0 &= \frac{c}{2l} \\ f_1 &= \frac{2c}{2l} \\ f_2 &= \frac{3c}{2l} \\ &\dots \end{aligned} \tag{8.18}$$

with for the case of a pre-stressed cable:

$$c = \sqrt{\frac{\sigma_0}{\rho}} \tag{8.19}$$

Substitution of $\sigma_0 = 500$ $[N]$, $l = 20$ $[m]$ and $\rho = 1.3636 * 10^4$ $[kg/m^3]$ yields:

$$\begin{aligned} f_0 &= 1.02 \ [Hz] \\ f_1 &= 2.04 \ [Hz] \\ f_2 &= 3.06 \ [Hz] \\ &\dots \end{aligned} \tag{8.20}$$

As can be seen from table 8.1, these results coincide within acceptable error margins.

Chapter 9

Transient analysis with cable element

The last part of the assignment is to activate the code for transient analysis of cable elements. As the macro processor for transient analysis **B2TRANS** has recently been rewritten and modified [13], the modified cable element can also be tested for transient analysis. For this purpose use is made of the same literature example as used in chapter 8 [4].

9.1 Transient analysis

Recall the discretized equations of motion eq.(8.1) presented in chapter 8, subsection 8.1.1. These equations can be solved by implicit time integration methods provided in **B2000** by the macro-processor **B2TRANS**.

The default method used is a linear multistep method algorithm devised by Park. A more detailed description of **B2TRANS**, the used solution methods and the required parameters can be found in ref. [13].

Recall from the previous chapters that the stiffness matrix becomes singular in quasi-static analysis, causing an unsolvable set of equations. In dynamic analysis however, the presence of the mass matrix compensates this. For example, when using the Jensen formulation (as is done in **B2TRANS**), the mass matrix and the stiffness matrix are combined resulting in the *dynamic stiffness matrix*. This matrix no longer becomes singular and can be factorized.

An important option with respect to quasi-static analysis with **B2CONT**, is that **B2TRANS** offers the possibility to restart the computation with the transient solver after running a quasi-static analysis with **B2CONT**. Obviously, the initial conditions (velocity, acceleration) in that case will be equal to zero for such a restart.

9.2 Example problems

9.2.1 Stretched cable submitted to transverse loading

Consider the example problem of a cable stretched between two horizontal supports and submitted to a transverse dynamic loading $p(t) = p_0 t$. See also ref. [4]. The cable has (stretched length) $L = 20$ $[m]$, 'prestress' $\sigma_0 = 500$ $[N]$, extensional rigidity $EA_0 = 2.2 * 10^6$ $[N]$ and mass per unit length $\rho_0 A_0 = 0.3$ $[kg/m]$. (See figure 9.1).

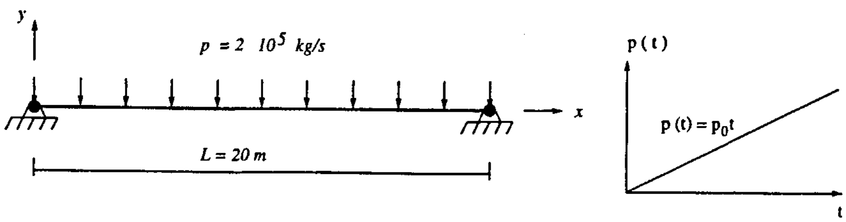
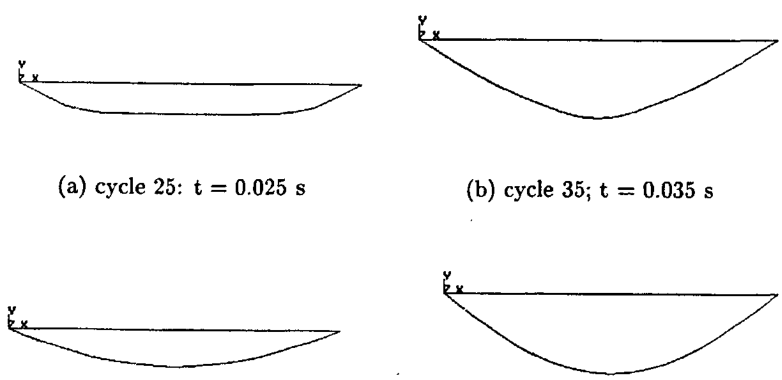


Figure 9.1: Stretched cable submitted to transverse loading

The cable is discretized into 20 elements and the dynamic load is applied with time increments of 0.001 $[s]$. The corresponding input-file can be found in appendix H. The prescribed end-time of 0.12 $[s]$ is reached after 120 time-steps. Figure 9.2 shows a sequence of displays of the cable configurations.



(a) cycle 25: t = 0.025 s

(b) cycle 35; t = 0.035 s

(c) cycle 55; t = 0.055 s

(d) cycle 75; t = 0.075 s

Figure 9.2: Dynamic response of cable submitted to transverse loading

The dynamic response of the cable is shown by the time-displacement diagram for the midspan-node (i.e. node 11). See figure 9.3. The transverse displacement v is plotted along the vertical axis and the time t in seconds along the horizontal axis .

The peak values obtained by this transient computation are compared with the values presented in ref. [4] (see table 9.1).

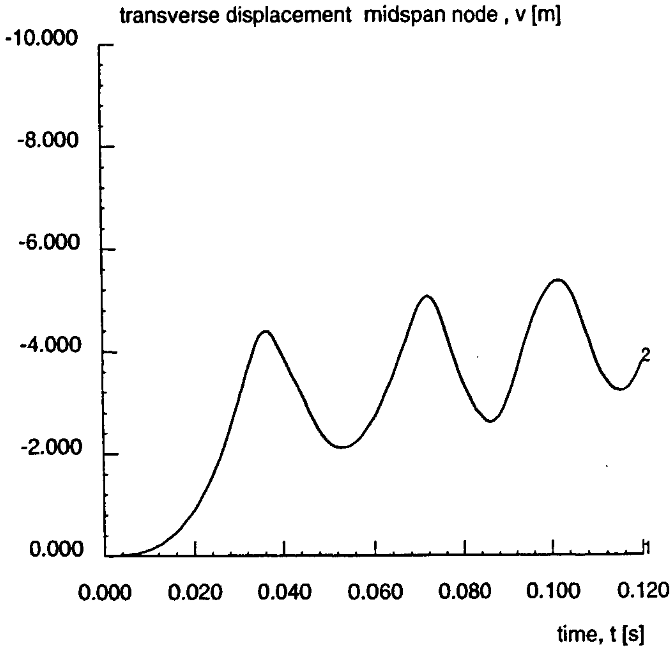


Figure 9.3: Transverse displacement versus time for the cable midspan node

The deviation in the numerical results are largest for the values of the transverse displacements. This deviation increases with increasing time. The deviation is probably caused by use of different solution techniques and a different form of discretization.

9.2.2 'Plucking' of guitar string

As a real life musical example consider a string of an acoustic guitar of scale length 648 $[mm]$. For the finite element model the length is taken to be 500 $[mm]$. The type of the string considered is **EJ15** Extra Light for $E - 1st$. The diameter D of the string is 0.0254 $[cm]$, yielding a cross sectional area A of $5.06707e - 02$ $[mm^2]$. The corresponding **prestress** $\sigma_0 * A$ is 72.0 $[N]$.

Furthermore, the string is assumed to be made of steel with Young's modulus $E = 0.2 * 10^12$ $[N/m^2]$ and density $\rho = 7.8 * 10^3$ $[kg/m^3]$. For the finite element model the string is discretized into 13 elements of 5.0 $[mm]$. See appendix H for the used input-file. The plucking of the string is simulated at $d = 15.0$ $[cm]$. (See figure 9.4).

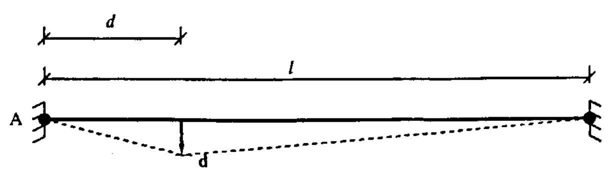


Figure 9.4: Plucking of an acoustic guitar string

	Geradin		*B2TRANS*		
Peak no.	*time* [s]	*displacement v* [m]	*time* [s]	*displacement v* [m]	*Error(v)*
1	0.036	-4.134	0.036	-4.406	0.066
2	0.069	-4.759	0.072	-5.081	0.068
3	0.098	-5.00	0.102	-5.378	0.076

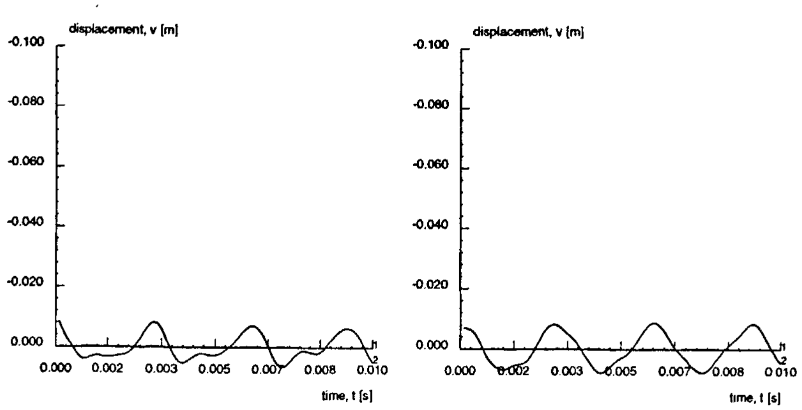
Table 9.1: Peak values obtained from resp. ref. [4] and **B2TRANS** computation

Recall from section 9.1 the possibility to restart a **B2CONT** computation with **B2TRANS**. This option makes it possible to simulate the plucking of the string as follows:

(1) The first part of the analysis consists of giving the string a small dispacement of 10.0 $[mm]$ at $d = 15.0$ $[mm]$ (node 4). This process can be simulated by quasi-static analysis (**B2CONT**), using prescribed displacement.

(2) In the second part of the analysis the string is released and the dynamic response is simulated. To achieve this, the computation is restarted from the last load-step (i.e. loadstep 11) from the static analysis, using ebv **B2TRANS**.

The dynamic responses of node 7 at distance 30 $[mm]$ from the point A and node 4 ('plucking' node) are given in figure 9.5.

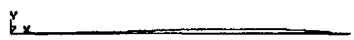


(a) node 4 (b) node 7

Figure 9.5: Time-displacement diagrams for respectively node 4 and node 7: E-1st

From the figure one can see that the displacement in the longitudinal direction (curve 1) is negligible, yielding only transverse modes. The corresponding frequency can be determined from the figure and is equal to 625 $[Hz]$. Figure 9.6 shows a sequence of displays of the cable configuration.

(a) cycle 11; t = 0.0 s



(b) cycle 25; t = 1.400E-03 s

(c) cycle 50; t = 3.900E-03 s

(d) cycle 100; t = 8.900E-03 s

Figure 9.6: Dynamic response of guitar-string due to 'plucking'

The Eigenfrequencies can be determined from linear vibration analysis (**B2LIN**), yielding for the lowest two Eigen frequencies 327.522 $[Hz]$ and 650.267 $[Hz]$.

Chapter 10

Conclusions and recommendations

10.1 Conclusions

The main objective of this assignment was to develop, implement and/or activate a geometrically nonlinear cable element which is suitable for applications in quasi-static analysis, linearized vibration analysis and transient (dynamic) analysis. As a first approximation the cable is assumed to behave like a string. This means that the element only has tangential stiffness. As a consequence the cable is completely unstable in the unloaded (tensionless) state, i.e. becomes singular. This aspect required an extensive evaluation particularly for quasi-static analysis, as this could happen in a variety of situations.

From the visits to the *National Aerospace Laboratory* or *NLR*, the interest for the (cable-supported) mast of a sailing yacht arised as an example of a practical application problem for the developed cable element with its implemented properties. Finding interpretations for this mast example and investigating modeling strategies became one of the sub-objectives of this assignment.

The singular behaviour of a tensionless cable can well be explained in the context of classical stability theory applied to a slender beam submitted to compression. A tensionless cable is completely unstable. This singular behaviour causes computational problems, specifically for quasi-static analysis (executed by running **B2CONT**), as it results in a singular stiffness matrix.

The original code presented a nonlinear cable element which did not satisfy the conditions required to enable some elementary applications involving tensionless cables. Also the formulations for the first and second variation for positive stress as well as the linearized stiffness matrix used for linear (vibration) analysis, required some corrections.

quasi-static analysis
Several methods to satisfy the specified conditions were investigated. These methods are all based on the principle to find ways to stabilize the cable such that continuation of the computation is assured. This can be achieved by either

defining a small positive strain in the cable or by defining a low modulus of elasticity E (Young's modulus):

* Treating the (tensionless) cable as an elastic rubber band by defining small values for the stiffness parameter (AE) enables quasi-static analysis of cable structures of which one or more cables become tensionless during the deformation process. These cables have negligible interference with the rest of the structure as they will behave like rubber elastics. However, this method does not enable computation with initially tensionless cables.

* The problem of initially tensionless cables was approached by giving the cable a small positive strain by re-definition of the initial (unstretched) length in the first load-cycle. This initializes the computation, after which the computation is continued in the conventional way with positive stress formulations. This method however does not provide the correct formulations for the condition described above.

Investigation of both methods has lead to the conclusion that an elegant solution strategy capturing both conditions could be extracted by proper combination of these two methods. This resulted in the *refined* l_0E*-method* as described in section 6.5. This method is implemented such that the user can specify the required method, i.e. the default method or the default method after initialization of the computation by the $l_0 - method$. The method offers solution possibilities for a broad variety of problems involving tensionless cables.

dynamic analysis: Linearized vibrations around stable states
The original formulation for the linearized stiffness matrix did not comprise the components responsible for transverse stiffness. Consequently, the effect of prestress was not present and no transverse Eigen modes were found. To activate the code for linearized vibrations around stable equilibrium states a correction of the second variation was required. The modified formulations were tested by Eigenmode analysis and verified analytically. It must be remarked that no 'negative stress' provisions are implemented for this type of analysis.

dynamic analysis: Transient analysis
Transient analysis offers a whole new perspective to cable applications. Singularities in the stiffness matrix are no longer problematic due to the presence of a mass matrix. From the test examples it was found that the developed cable element can be used for transient analysis without any problems.

An interesting option provided by **B2TRANS** is the possibility to restart with **B2TRANS** after a quasi-static analysis with **B2CONT**. This option was used in the example of the 'plucking of a guitar string' for the case of the '$E - 1st$.

10.2 Recommendations

Although the developed cable element appears to be applicable for a wide range of problems involving tensionless cables as well as stretched cables, some recommendations for further research are mentioned below;

- One recommendation can be made about fixing the deficiency in BASPL encountered in section 6.1. This deficiency results in an incorrect display of the deformation of the cable structure when no interior nodes are defined.

- As was explained in chapter 2, section 2.2, the cable can be interpreted as a beam with negligible bending stiffness. The effect of actually defining a (nonlinear) beam element with small bending stiffness instead of a cable can be investigated and compared to the results obtained by the cable element. Care must be taken to define the problems with the beam and the cable consistently and take into account that a prestressed cable has some transverse stiffness.

- The option to restart from a **B2CONT** analysis with a **B2TRANS** analysis also offers some further investigation perspectives. One can investigate the possibilities to restart with a **B2CONT** analysis after running **B2TRANS**. If this is possible it may be interesting to investigate another strategy to approach the singularity problems without actually modifying the first and second variations. The following procedure is also worth investigating: When in **B2CONT** the cable becomes tensionless the computation can be switched to a transient analysis. By implementation of a velocity and acceleration test the computation is switched back to a **B2CONT** analysis when the velocity and acceleration become sufficiently small. Recall that a singular stiffness matrix (of a tensionless cable) does not cause computation problems for transient analysis.

- Presently, the initial length and the stiffness parameter AE are adjusted for 'negative stress' cases by multiplying these values with a factor containing a *'small'* value (see *b2ep39.F*). This *'small'* value can be varied to investigate the minimum value by which the correct results are obtained.

- The *NLR* mast problem requires further investigation as well. More modeling strategies can be investigated as well as possibilities to include the initial cable lengths into the optimization procedure.

- An interesting problem that can be tested using **B2TRANS** is the pendulum problem, i.e. a cable fixed at the upper end with a mass connected to the lower end (node).

- As alternative formulations are used for the first and second variations in case of 'negative stress', also the transient solver **B2TRANS** uses these adjusted values. Although no problems are to be expected, it is recommended to investigate the effect of these modifications in transient analysis involving (temporary) tensionless cables.

Bibliography

[1] *PolyTechnisch zakboekje, A4/58-60.* Koninklijke PBNA b.v, Arnhem, 1997.

[2] D. O. Brush and B. O. Almroth. *Buckling of Bars, Plates and Shells.* McGraw-Hill Kogakusha, LTD, Tokyo, 1975.

[3] Y. C. Fung. *Foundations of solid mechanics.* Prentice-Hall, INC., Englewood Cliffs, New Jersey, 1965.

[4] M. Geradin, M. Hogge, and S. Idelsohn. Implicit finite element methods. In T. Belytschko and T. Hughes, editors, *Computational Methods for Transient Analysis*, volume 1 of *Computational Methods in Mechanics*, chapter 9, pages 417–471. North-Holland, Amsterdam, The Netherlands, 1983.

[5] W.T. Koiter. *Stijfheid en sterkte Grondslagen, mechanica 4.* Scheltema and Holkema N.V., Haarlem, 1972.

[6] A.S.K. Kwan. A new approach to geometric non-linearity of cable structures', computers and structures. *Computers and Structures, 67 (1998), 243-252*, 1998.

[7] J. L. Meek. *Computer methods in Structural analysis,first edition.* E & FN Spon, London, 1991.

[8] S. Merazzi. *B2000 Data Structure and Programming Handbook.* SMR Corporation, Bienne, Switzerland, 1994.

[9] S. Merazzi. *B2000 Processors Reference Manual, Version 1.8.* SMR Corporation, Bienne, Switzerland, 1994.

[10] L. Michielsen, H. de Vries, and P. Arendsen. Design of a composite mast for a yacht using B2000. *2nd B2000/MEMCOM Workshop at CSCS/SCSC*, November 5-6, 1998.

[11] National Aerospace Laboratory NLR. B2000, a modular finite element analysis system, development and applications.

[12] J. Przemieniecki. *Theory of matrix structural analysis.* Dover Publications, New York, 1968.

[13] J. J. C. Remmers. Mode-jumping with B2000, master's thesis. Master's thesis, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands, October, 1998.

[14] E. Riks. An incremental approach to the solution of snapping and buckling problems. *Int. J. Solids Structures, Vol 15, pp. 529-551*, 1978.

[15] E. Riks. Some computational aspects of the stability analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering, Vol 47, pp. 219-259*, 1984.

[16] N. Yoshitaka and M. Yasuyuki. A proposal for a cable element in finite element method. *Advances in Computational Engineering Science, 333-338.*

[17] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method, Fourth Edition, Volume 2, Solid and Fluid Mechanics, Dynamics and Non-linearity.* McGraw-Hill Book Company Europe, Berkshire, U.K., 1991.

[18] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method, Fourth Edition, Volume 1, Basic Formulation and Linear Problems.* McGraw-Hill Book Company Europe, Berkshire, U.K., 1994.

Appendix A

Original pre-variational element routine *b2epv39.F*

```
      subroutine b2epv39(coor, eprop,
     *                   epropall, elaminates,
     *                   elprev, work, irad, istat)
c
c   Compute prevariational data for 2-noded cables and rods
c   B2000  version 1.77
c
c   prev contains the following items:
c   prev(1)     - dx (initial length in x)
c   prev(2)     - dy
c   prev(3)     - dz
c   prev(4)     - initial cable/rod length. >0.0 cable_ident
c                                            <0.0 rod_ident
c   prev(5)     - initial cable cross section area
c
      implicit none
c
      real*8 coor(3,*)
      real*8 eprop(*),epropall(*)
      real*8 elaminates(4,*)
      real*8 elprev(*)
      real*8 work(*)
      integer irad(*)
      integer istat
c
#include "b2constants.ins"
#include "b2limits.ins"
#include "b2kernel.ins"
#include "b2test.ins"
c
      integer LSCONST
```

```
      parameter (LSCONST=20)
c       integer i

      real*8 sconst(LSCONST)
c
c 1100 format(' cable prevar=',5e15.7)
c
      istat=0
      elprev(1)=coorkern(1,2)-coorkern(1,1)
      elprev(2)=coorkern(2,2)-coorkern(2,1)
      elprev(3)=coorkern(3,2)-coorkern(3,1)
      elprev(4)=sqrt(elprev(1)**2 + elprev(2)**2 + elprev(3)**2)
c
c Compute section sconst (area)
c sconst = [ A Tconst Sy Sz Jy Jz Wy Wz ]
c here only A=sconst(1) is needed.

      call b2epropbsect(thkern,sconst,istat)
      if(istat.lt.0) return
      elprev(5)=sconst(1)

c   cable: compute initial_length for given prestress l = l0 ( 1 + p/ae )
c
      if(ecckern(1).gt.0.0) then
        elprev(4) = elprev(4) /
     *              ( 1.0 + ecckern(1)/(eprop(MATPOSE1)*sconst(1)) )

c   cable: initial_length given in eccentricty array ecckern

      else if(ecckern(1).lt.0.0) then
        elprev(4)=abs(ecckern(1))
c
c   rod: insert initial_length given and set to - to identify rod
c
      else
        elprev(4)=-elprev(4)
      endif
c
c      write(outkern,1100) (elprev(i),i=1,5)
      istat=0
      return
      end
```

Appendix B

Original element routine *b2ep39.F*

```
      subroutine b2ep39(coor, disp, etrans,
     *                  eprop, epropall, elaminates,
     *                  elprev, elurf, elfvar, elsvar, elstab,
     *                  eltfor,ellfor, plasold, plasnew,
     *                  work, irad, istat)
c
c B2000 subprogram to compute the first and the second variation of
c 2 noded cable and rod elements.
c
c B2000  version 1.8
c
c The output from b2epropbsect is
c
c     sconst = [ A Tconst Sy Sz Jy Jz Wy Wz ]
c
c modified 950315 sme section introduced
c modified 961104 sme b2epropbsect moved to b2epv. elprev(5)=Aread
c
      implicit none
c
c   Arguments
c
      real*8 coor(3,*), disp(*)
      real*8 etrans(*)
      real*8 eprop(*),epropall(*)
      real*8 elaminates(4,*)
      real*8 elprev(*),elurf(*),elfvar(*),elsvar(*),elstab(*)
      real*8 eltfor(*),ellfor(*), plasold(*), plasnew(*), work(*)
      integer irad(*)
      integer istat
c
```

```
#include "b2constants.ins"
#include "b2limits.ins"
#include "b2kernel.ins"
#include "b2io.ins"
#include "b2test.ins"
c
c   Local data
c
      integer LSCONST
      parameter (LSCONST=20)

      integer itens,irod,i,j

      real*8 zero,one,oldval
      real*8 rlen,dircos(3),rl0,ae,p,deltal,eps,delm1,scab(3,3)
      real*8 sconst(LSCONST)
c
      data zero/0.0/, one/1.0/
c
 1200 format(' cable var2 (scab)='/(3e12.4))
 1300 format('***WARNING Cable element ',i4,
     1        ' has negative prestress p=',1pe15.7)

c
c   *******************************************************************
c   linear? if so, compute prevariational data elprev
c   *******************************************************************
c
      if(dirkern(1).eq.0) then
        call b2epv39(coor, eprop, epropall, elaminates,
     *                elprev, work, irad, istat)
c print*,'elprev',(elprev(i),i=1,5)
        if(istat.lt.0) return
      endif
c
c   *******************************************************************
c   compute items common to first and second variation
c   *******************************************************************
c
c      if(ktest.gt.0) then
c        write(outkern,*) 'EPN11 cable el=',elikern
c        if(ktest.gt.2) then
c          write(outkern,*) 'elprev=',(elprev(i),i=1,4)
c          write(outkern,*) 'disp(1)=',(disp(i),i=1,3)
c          write(outkern,*) 'disp(2)=',(disp(i),i=4,6)
c        endif
c      endif
```

```
c
c  compute volume
c
      volkerno=abs(elprev(4)*elprev(5))
c
      rlen=zero
      do 10 i=1,3
      dircos(i)=elprev(i)+disp(i+3)-disp(i)
   10 rlen=rlen+dircos(i)**2
c
      if (rlen.eq.zero) then
        write(ioerr,'(A,I8)')
     *    ' ***ERROR (b2ep39): Length equal to zero. Element = ',elikern
        goto 900
      endif
      rlen=sqrt(rlen)
c
      do 20 i=1,3
   20 dircos(i)=dircos(i)/rlen
      rl0=abs(elprev(4))
      ae=eprop(MATPOSE1)*elprev(5)
c      if(ktest.gt.0) then
c        write(outkern,*) 'cur_length=',rlen,' ae=',ae
c        write(outkern,*) 'dircos=',(dircos(i),i=1,3)
c      endif
c
c  *********************************************************************
c  compute var2
c  *********************************************************************
c
      if(dirkern(3).eq.0) goto 100
c
c      if(ktest.gt.0) write(outkern,*) 'EPN11 compute VAR2'
c
c  geometric nonlinear
c
      if(dirkern(1).ne.0) then
        itens=0
c  check if rod element (elprev(4)<0.)
c
        if(elprev(4).lt.zero) then
          irod=1
        else
          irod=0    !! cable
          p=ae*(rlen-rl0)/rl0
          if(p.le.zero) itens=1
        endif
```

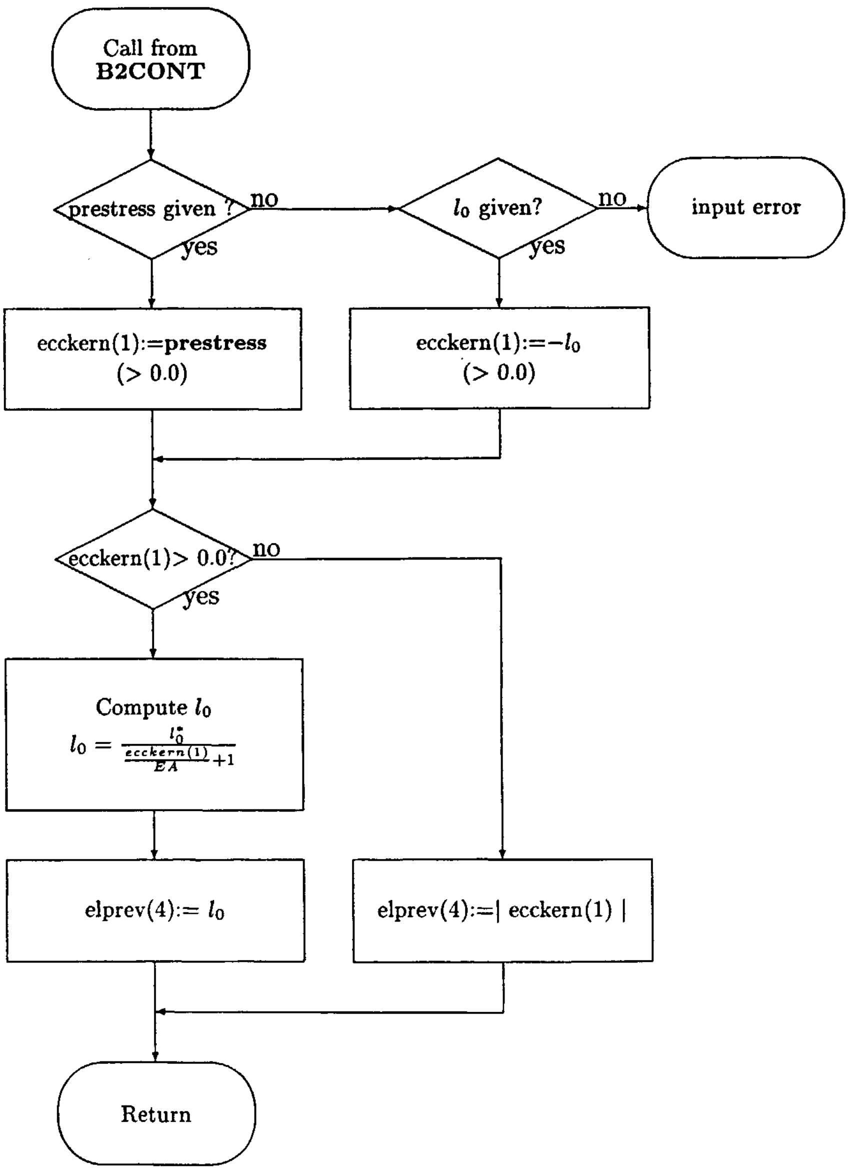
```
c
        deltal=(rlen-rl0)/rlen
        if(itens.eq.1.and.irod.eq.0) then
          deltal=zero
          write(outkern,1300) elikern,p
        endif
c         if(ktest.gt.0) write(outkern,*) 'rlen=',rlen,' deltal=',deltal
c
c   compute stress
c
      eps=(rlen-rl0)/rl0
c      if(matbkern.gt.0.and.abs(eps).gt.1.e-12) then
c        if(iprop(18).eq.4) then   !! deformation theory
c          call epplb4d(pltab,1,eps,sig)
c          ae=elprev(4)*abs(sig/eps)
c         print*,'**var2 ae=',ae
c        else
c          write(outkern,*) '**fatal** plasticity not impl., mthbv=',
c     1                   mtbhv
c          call mdump(sname,90)
c        endif
c      endif
c
        delm1=one-deltal
        do 60 i=1,3
   60   scab(i,i)= dircos(i)**2*delm1 +deltal
c
c   linear
c
      else
        delm1=one
        do 61 i=1,3
   61   scab(i,i)= dircos(i)**2
      endif
c
      ae=ae/rl0
      scab(1,2)=dircos(1)*dircos(2)*delm1
      scab(2,1)=scab(1,2)
      scab(1,3)=dircos(1)*dircos(3)*delm1
      scab(3,1)=scab(1,3)
      scab(2,3)=dircos(2)*dircos(3)*delm1
      scab(3,2)=scab(2,3)
      do 70 i=1,3
      do 70 j=1,3
   70 scab(i,j)=scab(i,j)*ae
c
c   the var2 matrix is stored in packed symmetric lower triangle.
```

```
c
      elsvar( 1)= scab(1,1)
c
      elsvar( 2)= scab(1,2)
      elsvar( 3)= scab(2,2)
c
      elsvar( 4)= scab(1,3)
      elsvar( 5)= scab(2,3)
      elsvar( 6)= scab(3,3)
c
      elsvar( 7)=-scab(1,1)
      elsvar( 8)=-scab(2,1)
      elsvar( 9)=-scab(3,1)
      elsvar(10)= scab(1,1)
c
      elsvar(11)=-scab(1,2)
      elsvar(12)=-scab(2,2)
      elsvar(13)=-scab(3,2)
      elsvar(14)= scab(1,2)
      elsvar(15)= scab(2,2)
c
      elsvar(16)=-scab(1,3)
      elsvar(17)=-scab(2,3)
      elsvar(18)=-scab(3,3)
      elsvar(19)= scab(1,3)
      elsvar(20)= scab(2,3)
      elsvar(21)= scab(3,3)
c
c       if(ktest.gt.2) write(outkern,1200) ((scab(j,i),j=1,3),i=1,3)
c
c   **********************************************************************
c   compute first variation
c   **********************************************************************
c
  100 if(dirkern(2).eq.0) goto 200
c
      eps=(rlen-rl0)/rl0
c
c   compute stress
c
c       if(matbkern.le.0) then
         ae=ae*eps
c       else
c         if(iprop(18).eq.4) then   !! deformation theory
c           call epplb4d(pltab,1,eps,sig)
c           a=elprev(4)*sig
c         else
```

```
c          write(outkern,*) '***FATAL plasticity not impl., mthbv=',
c     1                    matbkern
c          istat=-1
c          return
c        endif
c      endif
c
c   compute var1 (node forces)
c
      do 110 j=1,3
  110 dircos(j)=dircos(j)*ae
c      if(ktest.gt.1) write(outkern,*) 'var1=',(dircos(i),i=1,3)
      do 120 j=1,3
      elfvar(j)  =-dircos(j)
  120 elfvar(j+3)= dircos(j)
c
  200 istat=0
      return
c
  900 istat=-1
c
      end
```

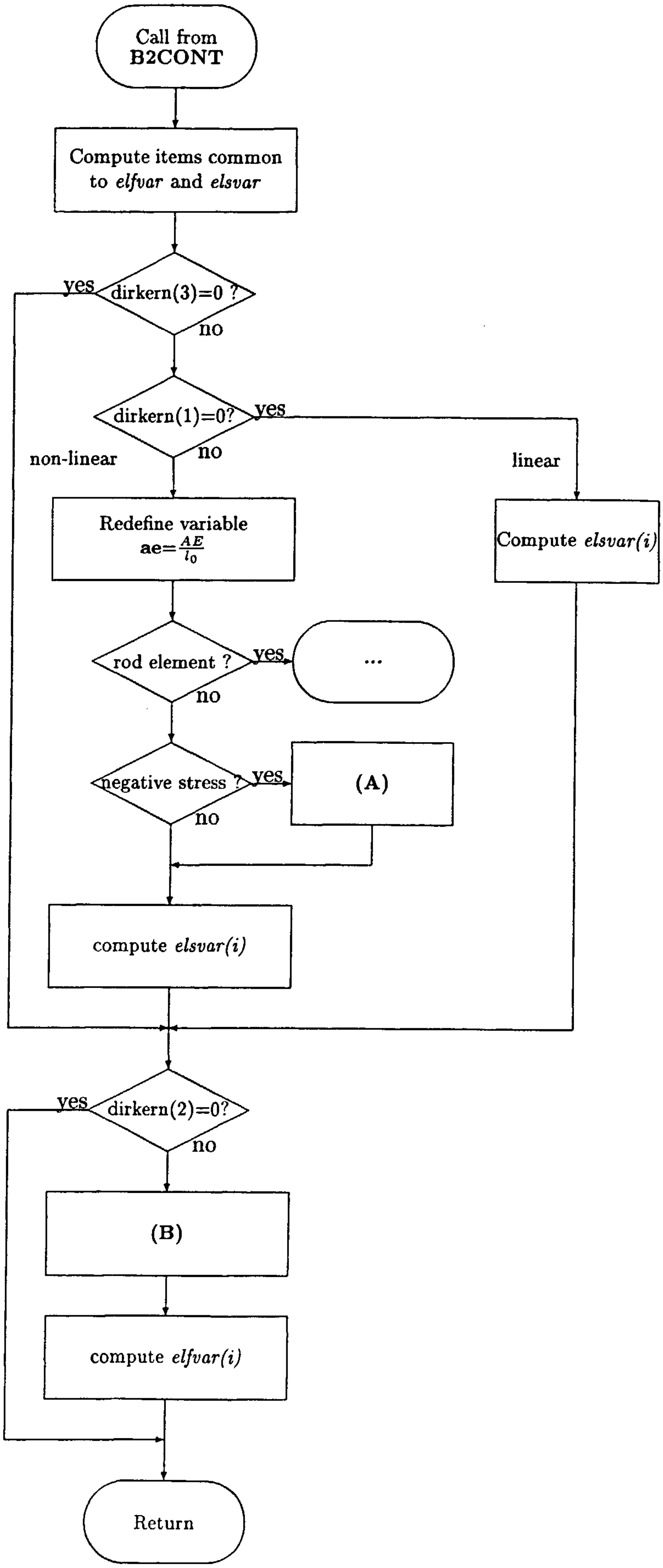
Appendix C

Flow diagram for *b2epv39.F*



Appendix D

Flow diagram for *b2ep39.F*


Call from
B2CONT
Compute items common to elfvar and elsvar
dirkern(3)=0 ?
yes
no
dirkern(1)=0?
yes
no
non-linear
linear
Redefine variable
ae=$\frac{AE}{l_0}$
Compute elsvar(i)
rod element ?
yes
no
...
negative stress ?
yes
no
(A)
compute elsvar(i)
dirkern(2)=0?
yes
no
(B)
compute elfvar(i)
Return

Appendix E

Modified element routines *b2epv39.F* and *b2ep39.F*

E.1 Modified *b2epv39.F*

```
      subroutine b2epv39(coor, eprop,
     *                   epropall, elaminates,
     *                   elprev, work, irad, istat)
c
c   Compute prevariational data for 2-noded cables and rods
c   B2000  version 1.77
c
c modified 981211 ps  Activated ecckern(2) and used to identify rod
c                     element. Negative values of ecckern(2) identify
c                     the 10-start method for the cable element,
c                     Positive values indicate default method. recognition
c                     of the requested method in b2ep39.F is established
c                     by using positive value of elprev(6) for the
c                     default method and negative value for the
c                     10-start method.
c
c   prev contains the following items:
c   prev(1)    - dx (initial length in x)
c   prev(2)    - dy
c   prev(3)    - dz
c   prev(4)    - initial cable/rod length. >0.0 cable_ident
c                                          <0.0 rod_ident
c   prev(5)    - initial cable cross section area
c-PS
c   prev(6)    - <0 for 10-method at start
c                >0 default method (10-E method)
c-PS
c
```

```
      implicit none
c
      real*8 coor(3,*)
      real*8 eprop(*),epropall(*)
      real*8 elaminates(4,*)
      real*8 elprev(*)
      real*8 work(*)
      integer irad(*)
      integer istat
c
#include "b2constants.ins"
#include "b2limits.ins"
#include "b2kernel.ins"
#include "b2test.ins"
c
      integer LSCONST
      parameter (LSCONST=20)
c      integer i

      real*8 sconst(LSCONST)
c
c 1100 format(' cable prevar=',5e15.7)
c
      istat=0
      elprev(1)=coorkern(1,2)-coorkern(1,1)
      elprev(2)=coorkern(2,2)-coorkern(2,1)
      elprev(3)=coorkern(3,2)-coorkern(3,1)
      elprev(4)=sqrt(elprev(1)**2 + elprev(2)**2 + elprev(3)**2)

c
c Compute section sconst (area)
c sconst = [ A Tconst Sy Sz Jy Jz Wy Wz ]
c here only A=sconst(1) is needed.

      call b2epropbsect(thkern,sconst,istat)
      if(istat.lt.0) return
      elprev(5)=sconst(1)

c-PS
c.....default method; set default value for elprev(6)

      elprev(6)=1.0
      if(ecckern(2).gt.0.5) then

c-PS
```

```
c   cable: compute initial_length for given prestress l = l0 ( 1 + p/ae )
c
      if(ecckern(1).gt.0.0) then
         elprev(4) = elprev(4) /
    *         ( 1.0 + ecckern(1)/(eprop(MATPOSE1)*sconst(1)) )

c   cable: initial_length given in eccentricity array ecckern

       else if(ecckern(1).lt.0.0) then
         elprev(4)=abs(ecckern(1))
c
c   rod: insert initial_length given and set to - to identify rod
c
c-PS         else
c-PS           elprev(4)=-elprev(4)
       endif

c-PS
c.....elprev(6) set to negative value for l0-method at start

      elseif(ecckern(2).lt.-0.5) then
        elprev(6)=-1.0

c-PS       else
c
c   rod: insert initial_length given and set to - to identify rod
c
      else
         elprev(4)=-elprev(4)
      endif
c-PS

c
c      write(outkern,1100) (elprev(i),i=1,6)
c-PS
c      write(errkern,*) 'ecckern(2) = ',ecckern(2)
c-PS
      istat=0
      return
      end
```

E.2 Modified *b2ep39.F*

```
      subroutine b2ep39(coor, disp, etrans,
     *                  eprop, epropall, elaminates,
     *                  elprev, elurf, elfvar, elsvar, elstab,
     *                  eltfor,ellfor, plasold, plasnew,
     *                  work, irad, istat)
c
c B2000 subprogram to compute the first and the second variation of
c 2 noded cable and rod elements.
c
c B2000  version 1.8
c
c The output from b2epropbsect is
c
c     sconst = [ A Tconst Sy Sz Jy Jz Wy Wz ]
c
c modified 950315 sme section introduced
c modified 961104 sme b2epropbsect moved to b2epv. elprev(5)=Aread
c modified 980327 ps  Value of ae restored to EA before first
c                     variation computation. This is vital if b2ep39
c                     is called with dirkern(3)=1 and dirkern(2)=1. In
c                     such a case ae is changed for second variation
c                     computation and the changed (incorrect) value is
c                     then used to compute the first variation without
c                     this bugfix.
c modified 980406 ps  write negative stress warning to screen
c modified 980501 ps  Negative stress check introduced prior to
c                     first variation computation. This is essential
c                     to avoid first variation computation with negative
c                     values of strain and to allow implementation
c                     of alternative formulation for such cases.
c modified 980613 ps  Implementation of 10-E method for negative
c                     stress case: adjustment of 10 and ae for
c                     first and second variation computation.
c modified 980731 ps  Correction second variation computation for
c                     linear analysis. This is necessary for
c                     correct computation of the tangential stiffness
c                     matrix at the origin used for linear analysis.
c modified 981211 ps  Option to start with the 10 method implemented,
c                     indicated by the value of elprev(6). Default method
c                     is 10-E method.
c

      implicit none
c
c   Arguments
```

```
c
      real*8 coor(3,*), disp(*)
      real*8 etrans(*)
      real*8 eprop(*),epropall(*)
      real*8 elaminates(4,*)
      real*8 elprev(*),elurf(*),elfvar(*),elsvar(*),elstab(*)
      real*8 eltfor(*),ellfor(*), plasold(*), plasnew(*), work(*)
      integer irad(*)
      integer istat
c
#include "b2constants.ins"
#include "b2limits.ins"
#include "b2kernel.ins"
#include "b2io.ins"
#include "b2test.ins"
c
c   Local data
c
      integer LSCONST
      parameter (LSCONST=20)

      integer itens,irod,i,j

      real*8 zero,one,oldval,small
      real*8 rlen,dircos(3),rl0,ae,p,deltal,eps,delm1,scab(3,3)
      real*8 sconst(LSCONST)
c
c-PS
c.....define 'small' for 10-E method purposes. Value can be
c.....changed according to personal notion.
c-PS
      data zero/0.0/, one/1.d+00/, small/1.d-4/
c
 1200 format(' cable var2 (scab)='/(3e12.4))
 1300 format('***WARNING Cable element ',i4,
     1       ' has negative prestress p=',1pe15.7)

c
c   ********************************************************************
c   linear? if so, compute prevariational data elprev
c   ********************************************************************
c
      if(dirkern(1).eq.0) then
        call b2epv39(coor, eprop, epropall, elaminates,
     *               elprev, work, irad, istat)
c print*,'elprev',(elprev(i),i=1,5)
        if(istat.lt.0) return
```

```
      endif
c
c   ******************************************************************
c   compute items common to first and second variation
c   ******************************************************************
c
c       if(ktest.gt.0) then
c         write(outkern,*) 'EPN11 cable el=',elikern
c         if(ktest.gt.2) then
c           write(outkern,*) 'elprev=',(elprev(i),i=1,4)
c           write(outkern,*) 'disp(1)=',(disp(i),i=1,3)
c           write(outkern,*) 'disp(2)=',(disp(i),i=4,6)
c         endif
c       endif
c
c   compute volume
c
      volkerno=abs(elprev(4)*elprev(5))
c
      rlen=zero
      do 10 i=1,3
      dircos(i)=elprev(i)+disp(i+3)-disp(i)
   10 rlen=rlen+dircos(i)**2
c
      if (rlen.eq.zero) then
        write(ioerr,'(A,I8)')
     *    ' ***ERROR (b2ep39): Length equal to zero. Element = ',elikern
        goto 900
      endif
      rlen=sqrt(rlen)
c
      do 20 i=1,3
   20 dircos(i)=dircos(i)/rlen
      rl0=abs(elprev(4))
      ae=eprop(MATPOSE1)*elprev(5)
c
c-PS
c.....print current values of rl0 and rlen
c
      write(errkern,*) 'init_length orig.=',rl0
      write(errkern,*) 'cur_length=',rlen
c-PS
c
c       if(ktest.gt.0) then
c         write(outkern,*) 'cur_length=',rlen,' ae=',ae
c         write(outkern,*) 'dircos=',(dircos(i),i=1,3)
c       endif
```

```
c
c    *********************************************************************
c    compute var2
c    *********************************************************************
c
       if(dirkern(3).eq.0) goto 100
c
c       if(ktest.gt.0) write(outkern,*) 'EPN11 compute VAR2'
c
c    geometric nonlinear
c
       if(dirkern(1).ne.0) then
         itens=0
c    check if rod element (elprev(4)<0.)
c
         write(errkern,*) 'elprev(4) = ',elprev(4)
         if(elprev(4).lt.zero) then
           irod=1
         else
           irod=0    !! cable
           p=ae*(rlen-rl0)/rl0
           if(p.le.zero) itens=1
c-PS
           write(errkern,*) 'p = ',p
c-PS
         endif
c
         deltal=(rlen-rl0)/rlen
c
c-PS
c.....New formulation for case of negative stress. (l0-method for start)
c.....When the deformed length rlen becomes smaller (or equal)
c.....to the original initial length rl0, the stress becomes negative.
c.....The initial length rl0 is then redefined such that the new value is
c.....slightly smaller than the deformed length rlen, hence conserving a
c.....small value for the stress in the cable at all times.
c
          if(itens.eq.1.and.irod.eq.0) then
c-PS           deltal=zero
c
c-PS
c.....write negative stress warning to screen
c-PS        write(outkern,1300) elikern,p
            write(errkern,1300) elikern,p
c

               rl0= (one-small)*rlen
```

```
c
c.....print current value of rl0
c
        write(errkern,*) 'init_length new=',rl0
c
c.....deltal for new value of rl0 (negative stress case)
c

          deltal=(rlen-rl0)/rlen

c-PS
c.....Assign small value to E (Young's modulus) in the
c.....case of negative stress...'elastic effect' (default)
c
          if(elprev(6).gt.0.0) then
            ae=small*ae
          endif
c-PS

c.....alternative
c-PS      deltal=small
        endif

c-PS
c.....end of changes to second variation computation

c
c        if(ktest.gt.0) write(outkern,*) 'rlen=',rlen,' deltal=',deltal
c
c   compute stress
c
c-PS      eps=(rlen-rl0)/rl0
c      if(matbkern.gt.0.and.abs(eps).gt.1.e-12) then
c        if(iprop(18).eq.4) then   !! deformation theory
c          call epplb4d(pltab,1,eps,sig)
c          ae=elprev(4)*abs(sig/eps)
c         print*,'**var2 ae=',ae
c        else
c          write(outkern,*) '**fatal** plasticity not impl., mthbv=',
c     1                   mtbhv
c          call mdump(sname,90)
c        endif
c      endif
c
       delm1=one-deltal
       do 60 i=1,3
   60  scab(i,i)= dircos(i)**2*delm1 +deltal
```

```
c
c   linear
c
c-PS
c.....Correction to definition linear stiffness matrix in origin:
c.....corresponds to tangential stiffness matrix in the origin.

      else
c-PS        delm1=one
         deltal=(rlen-rl0)/rlen
         delm1=one-deltal
         do 61 i=1,3
   61    scab(i,i)=dircos(i)**2*delm1+deltal
      endif
c-PS   61   scab(i,i)= dircos(i)**2
c-PS       endif
c-PS
c.....end of corrections to linear stiffness matrix
c
      ae=ae/rl0
      scab(1,2)=dircos(1)*dircos(2)*delm1
      scab(2,1)=scab(1,2)
      scab(1,3)=dircos(1)*dircos(3)*delm1
      scab(3,1)=scab(1,3)
      scab(2,3)=dircos(2)*dircos(3)*delm1
      scab(3,2)=scab(2,3)
      do 70 i=1,3
      do 70 j=1,3
   70 scab(i,j)=scab(i,j)*ae
c
c   the var2 matrix is stored in packed symmetric lower triangle.
c
      elsvar( 1)= scab(1,1)
c
      elsvar( 2)= scab(1,2)
      elsvar( 3)= scab(2,2)
c
      elsvar( 4)= scab(1,3)
      elsvar( 5)= scab(2,3)
      elsvar( 6)= scab(3,3)
c
      elsvar( 7)=-scab(1,1)
      elsvar( 8)=-scab(2,1)
      elsvar( 9)=-scab(3,1)
      elsvar(10)= scab(1,1)
c
      elsvar(11)=-scab(1,2)
```

```
      elsvar(12)=-scab(2,2)
      elsvar(13)=-scab(3,2)
      elsvar(14)= scab(1,2)
      elsvar(15)= scab(2,2)
c
      elsvar(16)=-scab(1,3)
      elsvar(17)=-scab(2,3)
      elsvar(18)=-scab(3,3)
      elsvar(19)= scab(1,3)
      elsvar(20)= scab(2,3)
      elsvar(21)= scab(3,3)
c
c      if(ktest.gt.2) write(outkern,1200) ((scab(j,i),j=1,3),i=1,3)
c
c   *********************************************************************
c   compute first variation
c   *********************************************************************
c
  100 if(dirkern(2).eq.0) goto 200
c
c-PS
c..... Restore values of rl0 (=10) and ae (=AE). This is needed
c......since a call to b2ep39 with dirkern(3)=1 will change the
c......values of ae and rl0 (in case of negative stress) to a
c......value that is incorrect for first variation computation.
c
      itens=0
      rl0=abs(elprev(4))
      ae=eprop(MATPOSE1)*elprev(5)
      eps=(rlen-rl0)/rl0
c-PS
c....print current value of rl0
c
      write(errkern,*) 'init_length orig.=',rl0
c
c   compute stress
c
c      if(matbkern.le.0) then
c
c-PS
c.....Check (again) for Rod element and negative stress
c.....prior to first variation computation
c
        if (elprev(4).lt.zero) then
          irod=1
        else
c-PS    if(elprev(4).ge.zero) then
```

```
          irod=0
          p=ae*eps
          if(p.le.zero) then
            itens=1
            write(errkern,1300) elikern,p

            rl0=(one-small)*rlen
c-PS
c.....Default method: superpose AE-method

            if(elprev(6).gt.0.0) then
              ae=small*ae
            endif

c-PS
            eps=(rlen-rl0)/rl0

c-PS
c.....end of rod and negative stress check

c.....print current value of rl0
      write(errkern,*) 'init_length new=',rl0
c
          endif
        endif
c
        ae=ae*eps
c      else
c        if(iprop(18).eq.4) then   !! deformation theory
c          call epplb4d(pltab,1,eps,sig)
c          a=elprev(4)*sig
c        else
c          write(outkern,*) '***FATAL plasticity not impl., mthbv=',
c     1                   matbkern
c          istat=-1
c          return
c        endif
c      endif
c
c   compute var1 (node forces)
c
      do 110 j=1,3
  110 dircos(j)=dircos(j)*ae
c      if(ktest.gt.1) write(outkern,*) 'var1=',(dircos(i),i=1,3)
      do 120 j=1,3
      elfvar(j)  =-dircos(j)
  120 elfvar(j+3)= dircos(j)
```

```
c
  200 istat=0

c-PS
c.....'reset' to default values

      if(elprev(6).lt.0.0) then
        elprev(6) = 1.0
        modpkerno=1
      endif
c-PS
      return
c
  900 istat=-1
c


      end
```

Appendix F

Input-files test-examples quasi-static analysis

F.1 UN-symmetric triangular cable contruction

```
#
# Triangular cable
#

 adir
   analysis collapse
   lca 1 pas 0.0 dpas 5.0 pamax 10.0
   lcb 2 pbs 5.0 dpbs 0.0 pbmax  5.0
   ncut 8  nfact 20 nstrat 0 maxit 15 maxstp 2
   epsdis 0.0000005
   epsr   0.0000005
   end
 branch 1
   btran
     trans  0.0 0.0 0.0
     rotx   0.0
     roty   0.0
     rotz  26.56505118
     end
   local
     lsid 1
       rotz   0.0
       endlsid
     lsid 2
       rotz -26.56505118
       endlsid
     end
   bdir
```

```
    deform NONLIN
    material LIN
    end
  nodes
    local 1
       1      0.0000000     0.0000000     0.0000000
       2     79.0450030     0.0000000     0.0000000
    local 2
       3    158.0900060     0.0000000     0.0000000
    end
  elem
    type c2 area 1.0 mid 1 prestress 3.7268
       1  1 2
       2  2 3
    end
  bound
    lock LLL 1
    lock FFL 2 3
    end
  force
    case 2
      type F dof 2 p 1.0000 3
    end
  endbranch
branch 2
  btran
    trans 141.4 70.7 0.0
    rotx    0.0
    roty    0.0
    rotz  -45.0
    end
  local
    lsid 1
      rotz   0.0
      endlsid
    lsid 2
      rotz 45.0
      endlsid
    end
  bdir
    deform NONLIN
    material LIN
    end
  nodes
    local 2
       1      0.0000000     0.0000000     0.0000000
    local 1
```

```
      2    50.0000000      0.0000000      0.0000000
      3   100.0000000      0.0000000      0.0000000
    end
  elem
    type c2 area 1.0 mid 1 prestress 4.7140
      1  1 2
      2  2 3
    end
  bound
    lock LLL 3
    lock FFL 1 2
    end
  force
    case 1
      type F dof 2 p 1.0 1
    end
  endbranch
join
  node 1 3 2 1
  end
emat
  mid 1
    type BEAM e 100.0 p 0.3
    endmid
  end
topology
renumber
run
```

F.2 Simple example problem to test prestress option

```
#
# Line cable
#

 adir
    analysis collapse
    lca 1 pas 0.0 dpas 0.2 pamax 0.4
    lcb 2 pbs 0.2 dpbs 0.0 pbmax 0.2
    ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 2
    epsdis 0.0001
    epsr   0.0001
 end
 branch 1
   bdir
     deform nonlinear
```

```
    end
  nodes
      1     0.0000000     0.0000000     0.0000000
      2     4.4000000     0.0000000     0.0000000
      3    10.7000000     0.0000000     0.0000000
    end
  elem
    type C2 area 2.0 mid 1 prestress 0.2
      1     1 2
    type C2 area 2.0 mid 1 prestress 0.1
      2     2 3
    end
  bound
    LOCK LLL 1
    LOCK FLL 2 3
    end
  force
    case 1
      type F dof 1 p 1.0 3
    case 2
      type F dof 1 p 0.5 2 3
    end
endbranch
emat
  mid 1
    type BEAM e 1.0 p 0.3
    endmid
  end
run
```

To test the **l0** option replace **element** definition part by:

```
  elem
    type C2 area 2.0 mid 1 l0 4.0
      1     1 2
    type C2 area 2.0 mid 1 l0 6.0
      2     2 3
    end
```

To test the prescribed displacements option replace the **force** definition part by:

```
  force
    case 1
      type D dof 1 p 5.0 3
    case 2
      type F dof 1 p 0.5 2 3
    end
```

F.3 Simple cable sag problem

```
#
# Line cable sag
#

 adir
    analysis collapse
    lca 1 pas 0.0 dpas 1.0 pamax 10
    lcb 0
    ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 10
    epsdis 0.0001
    epsr   0.0001
 end
 branch 1
   bdir
     deform nonlinear
     end
   nodes
       1     0.0000000     0.0000000     0.0000000
       2     1.1000000     0.0000000     0.0000000
       3     2.2000000     0.0000000     0.0000000
       4     3.3000000     0.0000000     0.0000000
       5     4.4000000     0.0000000     0.0000000
     end
   elem
     type C2 area 0.0001 mid 1 10 1.0
       1     1 2
       2     2 3
       3     3 4
       4     4 5
     end
   bound
     LOCK LLL 1
     LOCK FLL 5
     LOCK FFL 2 3 4
     end
   force
     case 1
       type F dof 2 p -1.0 2 3 4
       type D dof 1 p -0.04 5
     end
 endbranch
 emat
   mid 1
     type BEAM e 1.e6 p 0.3
     endmid
```

```
  end
 run
```

F.4 Cable as part of construction

```
#
# Triangular cable: one midnode per cable (/branch)
#

 adir
   analysis collapse
   lca 1 pas 0.0 dpas 5.0 pamax 10.0
   lcb 2 pbs 5.0 dpbs 0.0 pbmax  5.0
   ncut 8  nfact 20 nstrat 0 maxit 15 maxstp 2
   epsdis 0.0000005
   epsr   0.0000005
   end

 branch 1
   btran
     trans  0.0 0.0 0.0
     rotx   0.0
     roty   0.0
     rotz  45.0
     end
   local
     lsid 1
       rotz   0.0
       endlsid
     end
   bdir
     deform NONLIN
     material LIN
     end
   nodes
     local 1
        1      0.0000000      0.0000000      0.0000000
        2     50.0000000      0.0000000      0.0000000
        3    100.0000000      0.0000000      0.0000000
     end
   elem
     type c2 area 1.0 mid 1 prestress 5.0
        1  1 2
        2  2 3
     end
```

```
  bound
    lock LLL 1
    lock FFL 2 3
    end
  force
    case 2
      type F dof 1 p 1.0000 3
    end
  endbranch
branch 2
  btran
    trans  70.71067812  70.71067812 0.0
    rotx    0.0
    roty    0.0
    rotz  -45.0
    end
  local
    lsid 1
      rotz   0.0
      endlsid
    lsid 2
      rotz 45.0
      endlsid
    end
  bdir
    deform NONLIN
    material LIN
    end
  nodes
    local 2
       1      0.0000000     0.0000000     0.0000000
    local 1
       2     50.0000000     0.0000000     0.0000000
       3    100.0000000     0.0000000     0.0000000
    end
  elem
    type c2 area 1.0 mid 1 prestress 5.0
       1  1 2
       2  2 3
    end
  bound
    lock LLL 3
    lock FFL 1 2
    end
  force
    case 1
      type F dof 1 p 1.0 1
```

```
     end
   endbranch
 branch 3
   btran
     trans  0.0 141.4213562  0.0
     rotx    0.0
     roty    0.0
     rotz  -45.0
     end
   local
     lsid 1
       rotz   0.0
       endlsid
     lsid 2
       rotz 45.0
       endlsid
     end
   bdir
     deform NONLIN
     material LIN
     end
   nodes
     local 1
        1      0.0000000      0.0000000      0.0000000
        2     50.0000000      0.0000000      0.0000000
        3    100.0000000      0.0000000      0.0000000
     end
   elem
     type c2 area 1.0 mid 1 prestress 5.0
        1  1 2
        2  2 3
     end
   bound
     lock FFL 2 3
     lock LLL 1
     end
   endbranch
join
  node 1 3 2 1
  node 1 3 3 3
  end
emat
  mid 1
    type BEAM e 100.0 p 0.3
    endmid
  end
topology
```

```
renumber
run
```

F.5 Direct computation of cable sag; start problem

```
#
# Line cable sag
#
 adir
    analysis collapse
    lca 1 pas 0.0 dpas 1.0 pamax 10
    lcb 0
    ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 10
    epsdis 0.0001
    epsr   0.0001
 end
 branch 1
   bdir
     deform nonlinear
     end
# nodes corresponding to initial length 10
   nodes
       1      0.0000000      0.0000000      0.0000000
       2      1.0000000      0.0000000      0.0000000
       3      2.0000000      0.0000000      0.0000000
       4      3.0000000      0.0000000      0.0000000
       5      4.0000000      0.0000000      0.0000000
     end
   elem
     type C2 area 0.0001 mid 1 10 1.0
       1      1 2
       2      2 3
       3      3 4
       4      4 5
     end
   bound
     LOCK LLL 1
# no prescribed displacements, such that node 5 must be locked
# now in all directions
     LOCK LLL 5
     LOCK FFL 2 3 4
     end
   force
     case 1
```

```
      type F dof 2 p -1.0 2 3 4
    end
endbranch
emat
  mid 1
    type BEAM e 1.e6 p 0.3
    endmid
  end
run
```

F.6 Sagged equilibrium state; part 1

```
#
# Triangular cable , g-method part 1
#

adir
  analysis collapse
  lca 1 pas 0.0 dpas 1.0 pamax 10.0
  ncut 8  nfact 20 nstrat 0 maxit 50 maxstp 10
  epsdis 0.0000005
  epsr   0.0000005
  end
branch 1
  btran
    trans  0.0 0.0 0.0
    rotx   0.0
    roty   0.0
    rotz  45.0
    end
  local
    lsid 1
      rotz   0.0
      endlsid
    lsid 2
      rotz -45.0
      endlsid
    end
  bdir
    deform NONLIN
    material LIN
    end
nodes
  local 2
```

```
        1    0.0000000        0.0000000       0.0000000
        2    25.0000000       0.0000000       0.0000000
        3    50.0000000       0.0000000       0.0000000
        4    75.0000000       0.0000000       0.0000000
     local 2
        5  100.0000000        0.0000000       0.0000000
      end
    elem
      type c2 area 1.0 mid 1 10 24.13230876
         1  1 2
         2  2 3
         3  3 4
         4  4 5
      end
    bound
      lock LLL 1
      lock FFL 2 3 4 5
      end
    force
      case 1
        type F dof 2 p -0.01000 2 3 4
        type D dof 2 p -0.5000 5
      end
    endbranch
  branch 2
    btran
      trans  70.71067812  70.71067812 0.0
      rotx    0.0
      roty    0.0
      rotz  -45.0
      end
    local
      lsid 1
        rotz   0.0
        endlsid
      lsid 2
        rotz 45.0
        endlsid
      end
    bdir
      deform NONLIN
      material LIN
      end
    nodes
     local 2
         1     0.0000000      0.0000000      0.0000000
      local 2
```

```
      2    25.0000000     0.0000000     0.0000000
      3    50.0000000     0.0000000     0.0000000
      4    75.0000000     0.0000000     0.0000000
      5   100.0000000     0.0000000     0.0000000
    end
  elem
    type c2 area 1.0 mid 1 10 24.13230876
      1  1 2
      2  2 3
      3  3 4
      4  4 5
    end
  bound
    lock LLL 5
    lock FFL  1 2 3 4
    end
  force
    case 1
      type F dof 2 p -0.01000 2 3 4
    end
  endbranch
join
  node 1 5 2 1
  end
emat
  mid 1
    type BEAM e 100.0 p 0.3
    endmid
  end
topology
renumber
run
```

F.7 Sagged equilibrium state; part 2

```
#
# Triangular cable   , g-method part 2
#

 adir
   analysis collapse
   lca 1 pas 0.0 dpas 5.0 pamax 10.0
   lcb 2 pbs 1.0 dpbs 0.0 pbmax  1.0
   ncut 8  nfact 20 nstrat 0 maxit 50 maxstp 10
```

```
  epsdis 0.0000005
  epsr   0.0000005
  end
branch 1
  btran
    trans  0.0 0.0 0.0
    rotx   0.0
    roty   0.0
    rotz  45.0
    end
  local
    lsid 1
      rotz   0.0
      endlsid
    lsid 2
      rotz -45.0
      endlsid
    end
  bdir
    deform NONLIN
    material LIN
    end
  nodes
    local 2
      1     0.0000000      0.00000000      0.0000000
      2    23.7689000     -4.9446E+00      0.0000000
      3    47.9862000     -6.8513E+00      0.0000000
      4    72.2871000     -6.2303E+00      0.0000000
    local 2
      5    96.4645000     -3.5355E+00    0.0000000
    end
  elem
    type c2 area 1.0 mid 1 10 24.13230876
      1  1 2
      2  2 3
      3  3 4
      4  4 5
    end
  bound
    lock LLL 1
    lock FFL 2 3 4 5
    end
  force
    case 2
      type F dof 2 p -0.1000 2 3 4
      type F dof 2 p 1.260785533 5
    end
```

```
  endbranch
branch 2
  btran
    trans  70.71067812  70.71067812 0.0
    rotx    0.0
    roty    0.0
    rotz  -45.0
    end
  local
    lsid 1
      rotz  -22.5
     endlsid
    lsid 2
     rotz 45.0
     endlsid
    end
  bdir
    deform NONLIN
    material LIN
    end
  nodes
   local 1
     1    3.5355E+00     -3.5355E+00      0.0000000
    local 2
     2   27.7129E+00     -6.2303E+00      0.0000000
     3   52.0138E+00     -6.8513E+00      0.0000000
     4   76.2311E+00     -4.9446E+00      0.0000000
     5  100.0000000       0.00000000      0.0000000
    end
  elem
    type c2 area 1.0 mid 1 10 24.13230876
      1  1 2
      2  2 3
      3  3 4
      4  4 5
    end
  bound
    lock LLL 5
    lock FFL  1 2 3 4
    end
  force
    case 1
      type F dof 2 p 1.0 1
    case 2
      type F dof 2 p -0.100 2 3 4
    end
  endbranch
```

```
join
  node 1 5 2 1
  end
emat
  mid 1
    type BEAM e 100.0 p 0.3
    endmid
  end
topology
renumber
run
```

Corresponding scriptfiles for quasi-static **B2CONT** analysis:

```
cp a b
b2cont<< /
ar a
co b
test 0
extrapolate 3
newmod 1
pathpar 0
fullnewton
even
felippa
output b2cont.out
go
```

Used **B2CONT** *restart* scriptfile :

```
#
b2cont << /
ar a
co b
adir
step    2
currcy  2
maxstp 50
nfact  200
pamax  995.0
epsdis 0.0000005
epsr   0.0000005
end
restartfac 1.0
newmod 1
pathpar 0
```

```
lutol 1.0e-09
lutol 1.0e-09
fullnewton
extrapolate 3
output b2contre.out
felippa
go
```

Appendix G

Input-file test-example Eigenmode analysis

```
adir
  lca 1 pas 0.0 dpas 0.01 pamax 1.0
  ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 250
  epsdis 0.0001
  epsr   0.0001
end

branch 1
  bdir
    MATERIAL ELASTIC
    deform nonlinear
    end
  nodes
       1      0.0000000      0.0000000      0.0000000
       2      1.0000000      0.0000000      0.0000000
       3      2.0000000      0.0000000      0.0000000
       4      3.0000000      0.0000000      0.0000000
       5      4.0000000      0.0000000      0.0000000
       6      5.0000000      0.0000000      0.0000000
       7      6.0000000      0.0000000      0.0000000
       8      7.0000000      0.0000000      0.0000000
       9      8.0000000      0.0000000      0.0000000
      10      9.0000000      0.0000000      0.0000000
      11     10.0000000      0.0000000      0.0000000
      12     11.0000000      0.0000000      0.0000000
      13     12.0000000      0.0000000      0.0000000
      14     13.0000000      0.0000000      0.0000000
      15     14.0000000      0.0000000      0.0000000
      16     15.0000000      0.0000000      0.0000000
      17     16.0000000      0.0000000      0.0000000
      18     17.0000000      0.0000000      0.0000000
```

```
      19     18.0000000      0.0000000      0.0000000
      20     19.0000000      0.0000000      0.0000000
      21     20.0000000      0.0000000      0.0000000
     end
   elem
    type C2 area 2.2e-05 mid 1 prestress 500
       1      1 2
       2      2 3
       3      3 4
       4      4 5
       5      5 6
       6      6 7
       7      7 8
       8      8 9
       9      9 10
      10     10 11
      11     11 12
      12     12 13
      13     13 14
      14     14 15
      15     15 16
      16     16 17
      17     17 18
      18     18 19
      19     19 20
      20     20 21
     end
    MASS
     TYPE CO
     ELEMENTS 1/20
     END
  bound
     LOCK LLL 1 21
     LOCK FFL 2/20
     end
 force
     case 1
     DOF 2 P -2.0 2/20
     end
endbranch
emat
  mid 1
    type BEAM e 1.0e11 p 0.3
    DENS 1.36364e04
    endmid
  end
run
```

Appendix H

Input-files test-examples transient analysis

H.1 Stretched cable submitted to transverse loading

```
#
# Stretched cable submitted to transverse loading
#

adir
   analysis nonlinear
   lca 0
   lcb 0
   ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 240
   epsdis 0.0001
   epsr   0.0001
 end

DYNA
   TIME_S 0.0
   TIME_E 0.12
   DT 0.001
   LOAD 1 SLOPE 0. 0.12  24000.0
END
 branch 1
   bdir
     MATERIAL ELASTIC
     deform nonlinear
     end
   nodes
       1      0.0000000      0.0000000      0.0000000
       2      1.0000000      0.0000000      0.0000000
       3      2.0000000      0.0000000      0.0000000
```

```
    4      3.0000000     0.0000000     0.0000000
    5      4.0000000     0.0000000     0.0000000
    6      5.0000000     0.0000000     0.0000000
    7      6.0000000     0.0000000     0.0000000
    8      7.0000000     0.0000000     0.0000000
    9      8.0000000     0.0000000     0.0000000
   10      9.0000000     0.0000000     0.0000000
   11     10.0000000     0.0000000     0.0000000
   12     11.0000000     0.0000000     0.0000000
   13     12.0000000     0.0000000     0.0000000
   14     13.0000000     0.0000000     0.0000000
   15     14.0000000     0.0000000     0.0000000
   16     15.0000000     0.0000000     0.0000000
   17     16.0000000     0.0000000     0.0000000
   18     17.0000000     0.0000000     0.0000000
   19     18.0000000     0.0000000     0.0000000
   20     19.0000000     0.0000000     0.0000000
   21     20.0000000     0.0000000     0.0000000
  end
elem
  type C2 area 2.2e-05 mid 1 p 500
    1      1 2
    2      2 3
    3      3 4
    4      4 5
    5      5 6
    6      6 7
    7      7 8
    8      8 9
    9      9 10
   10     10 11
   11     11 12
   12     12 13
   13     13 14
   14     14 15
   15     15 16
   16     16 17
   17     17 18
   18     18 19
   19     19 20
   20     20 21
  end
MASS
  TYPE CO
  ELEMENTS 1/20
  END
bound
```

```
    LOCK LLL 1 21
    LOCK FFL 2/20
    end
  force
    case 1
    DOF 2 P -1.0 2/20
    end
endbranch
emat
  mid 1
    type BEAM e 1.0e11 p 0.3
    DENS 1.36364e04
    endmid
  end
run
```

H.2 'Plucking' of an acoustic guitar string

```
#
# Guitar string
#
adir
   analysis nonlinear
# define in case 2 as B2TRANS only reads case 1
# (no load for B2TRANS computation)
   lca 2 pas 0.0 dpas 0.1 pamax 1.0
   lcb 0
   ncut 8  nfact 2000 nstrat 0 maxit 15 maxstp 240
   epsdis 0.0001
   epsr   0.0001
 end

 DYNA
    TIME_S 0.0
    TIME_E 0.01
    DT 0.0001
    INIT_DISP
END
 branch 1
   bdir
     MATERIAL ELASTIC
     deform nonlinear
     end
   nodes
       1      0.0000000      0.0000000      0.0000000
```

```
        2      0.0500000     0.0000000     0.0000000
        3      0.1000000     0.0000000     0.0000000
        4      0.1500000     0.0000000     0.0000000
        5      0.2000000     0.0000000     0.0000000
        6      0.2500000     0.0000000     0.0000000
        7      0.3000000     0.0000000     0.0000000
        8      0.3500000     0.0000000     0.0000000
        9      0.4000000     0.0000000     0.0000000
       10      0.4500000     0.0000000     0.0000000
       11      0.5000000     0.0000000     0.0000000
       12      0.5500000     0.0000000     0.0000000
       13      0.6000000     0.0000000     0.0000000
       14      0.6500000     0.0000000     0.0000000
      end
    elem
      type C2 area 5.06707e-08 mid 1 prestress 72.0
        1      1 2
        2      2 3
        3      3 4
        4      4 5
        5      5 6
        6      6 7
        7      7 8
        8      8 9
        9      9 10
       10     10 11
       11     11 12
       12     12 13
       13     13 14
      end
    MASS
      TYPE CO
      ELEMENTS 1/13
      END
    bound
      LOCK LLL 1 14
      LOCK FFL 2/13
      end
    force
      case 2
          type D DOF 2 P -0.01 4
      end
endbranch
emat
  mid 1
    type BEAM e 0.2e12 p 0.3
    DENS 7.8e3
```

```
    endmid
  end
run
```

Used *PCL* commands for transient analysis (**B2TRANS**):

```
>ar a
>co b
>adir
>step 11
>currcy 11
>end
>go
```

Appendix I

Element mass routine *b2mp39.F*

```
c-PS
c   subroutine copied from b2mp35.F (rod-2 element).
c-PS
c
      subroutine b2mp39(xyz, eprop, epropall, elaminates, elprev,
     *                  sme, work, irad, istat)
c
c   Compute element mass, cable-2 element
c
c
c   91-02-07 Transformation of local el-mass to global
c            mass matrix added.
c   ADB
c
c   bmtypkern  - type of mass gen. =ld lumped diag.
c                                  =cd consistent diag.
c                                  =co consistent
c   nodkern            - el. nodes
c   nnekern            - n. of nodes
c   coorkern           - branch cartesian coord.
c   eprop(MATPOSDENS) - mass density
c   thkern             - thkern(2)=area [m2]
c                        thkern(3)=polar inertia moment [m4]
c   sme                - mass vector/matrix (o)
c   istat              - status (o)
c                         =0 ok
c                         >0 cannot compute
c                         <0 ko
c
      implicit none
```

```
c
#include "b2limits.ins"
#include "b2constants.ins"
#include "b2kernel.ins"
#include "b2test.ins"
c
      integer irad(*), istat
      real*8 xyz(3,*), eprop(*), epropall(*)
      real*8 elaminates(4,*)
      real*8 elprev(*), sme(*), work(*)
c
      integer i
      real*8  a,al,alen
      real*8  zero,five,etol
c
      integer LSCONST
      parameter (LSCONST=20)
c
      real*8 sconst(LSCONST)
c
      data zero/0.0/, five/0.5/, etol/1.d-20/
c
 1000 format(' ***MP-ERROR*** Rod element has length 0. el=',i6)
c
      istat=0

c  Compute section sconst (area)
c    sconst = [ A Tconst Sy Sz Jy Jz Wy Wz ]
c    here only A=sconst(1) is needed.

      call b2epropbsect(thkern,sconst,istat)
      if(istat.lt.0) return
c
      alen=0.
c
c  Check length (alen) of element
c
      do 5 i=1,3
        alen=alen+(coorkern(i,2)-coorkern(i,1))**2
    5 continue
c
      alen=sqrt(alen)
      if(alen.lt.etol) then
        write(errkern,1000) elikern
        istat=-1
        return
      endif
```

```
c
c  Calculate mass of element and divide it over the 2 nodes
c     m(node) = .5*length*dens*area
c
      al=alen*eprop(MATPOSDENS)
      a=five*sconst(1)*al
      masskerno=sconst(1)*al
c
      if(bmtypkern.eq.BMTYPCD.or.bmtypkern.eq.BMTYPLD) then
        call b2setfloat(zero,sme,6)
        do 11 i=1,3
          sme(i)=a
          sme(i+3)=a
 11     continue
      elseif(bmtypkern.eq.BMTYPCO) then
c
c  Simulate consistent mass by lumped diagonal
c
        call b2setfloat(zero,sme,21)
        sme(1) =a
        sme(3) =a
        sme(6) =a
        sme(10)=a
        sme(15)=a
        sme(21)=a
      endif
c
      return
      end
```