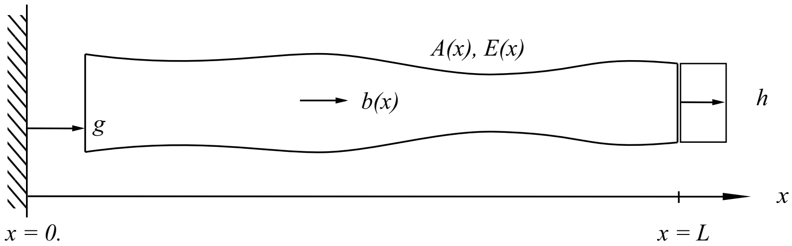
Finite Elements

"from the early beginning to the very end"



An Introduction to Elasticity and Heat Transfer Applications

Only the Bar and the Beam chapters!

Preliminary edition

LiU-IEI-S--08/535--SE

Bo Torstenfelt

Linköpings universitet
TEKNISKA HÖGSKOLAN

Contents

Preface

The writing of this book has arisen as a natural next step in my profession as a teacher, researcher, program developer, and user of the Finite Element Method. Of course one can wonder, why I am writing just another book in Finite Elements. The answer is equally obvious as simple. After many years in the field I have, as have many others, discovered a large variety of pitfalls or mistake done by others and myself. I have now reached a point where I would like to describe *my view of the topic*. That is, how to understand it, how to teach it, how to implement it and how to use it; these will be main goals for the discussion to come!

The discussion to come will be influenced by experiences from all four of these branches. As a teacher I have taught both basic and advanced courses in Finite Elements with focus on both Solid Mechanics and Heat Transfer applications. These courses have been given at the Linköping University in the mechanical engineering programme.

The text to come is written by an engineer for engineers. One overall goal for the description is to try to cover every step from how a certain mathematical model appears from basic considerations based on fact from reality, to a classical formulation with its possible analytical solutions, and finally over to a study of numerical solutions used by a Finite Element program based on a certain finite element formulation.

That is, discussing the finite element method *from the early beginning to the very end.*

The great challenge is to make this as short and interesting as possible without loosing or breaking the mathematical chain. It is strongly believed that for success in learning Finite Elements it is an absolute prerequisite to be familiar with the local equations and their available analytical solutions. I think most people who have tried to teach Finite Elements agree upon this, traditionally however, most education in Finite Elements is given in separate courses. Why not try to teach Finite Elements in close connection to where the basic material is taught. That is, integrate Finite Elements together with basic material in the same course!

Of course, Finite Elements can be taught as a weighted residual method for approximate solutions of sets of coupled partial differential equations without discussing any physical application and just focusing on existence, uniqueness and error bounds of the solution. This is of course also important but for most students studying different engineering disciplines Finite Elements will be a tool for trying to understand and predict the behavior of reality. An important focus in studies of finite element formulations of different engineering disciplines is to be aware what should be expected of the quality of the approximate numerical solution. This can only be learnt by knowing the most important details of the mathematical background and then solving numerical problems having an analytical solution to compare with.

Another aspect important for engineers working with the method as a daily tool, is how to use the method as efficiently as possible both from a time-consuming and a computer resources point of view.

Over a period of at least 15 years I have worked with the graphical finite element environment **TRINITAS**. This is a stand-alone tool for optimization, conceptual design and education as well as for general linear elasticity and heat transfer problem both as steady-state, transient or as eigenvalue problems. It is an Object Oriented program based on a graphical user interface for manipulation of the database of the program. This program contains procedures for geometry modeling, domain property and boundary conditions definition, mesh generation, finite element analysis and result evaluation. The program is used in eduction at different levels. It is used in basic courses in Finite Elements at an undergraduate level and also in advanced course where the students add their own routines for instance; element stiffness matrix, stress calculations in elasticity problems or utilizing ready-to-use routines for crack propagation analysis. This finite element environment has also been used for testing of different research ideas and for solving of different industrial engineering applications.

This program will be used throughout the text in this book as a tool for analysis of all examples given during the discussion of different finite element applications.

The ideas and the arguments given above have been the main driving force for doing this work. Hopefully, this book will prove useful as both an introduction of the method and also a standard tool or companion to be used during daily finite element work.

Bo Torstenfelt

November, 2007

Reader's Instruction

Readers that have never studied Finite Elements are recommended to first read the bar chapter (chap. 3) "*from the early beginning to the very end*" very carefully. It is the author's belief that this chapter is detailed enough to serve as a stand-alone base for self-studies where the reader is recommended to, during reading the text, perform a complete rewriting of the basic mathematical chain.

Every chapter to come is written in a manner and with an aim to be more or less self-contained for the reader with sufficient pre-qualifications. Typical required qualifications are 2 years studies at undergraduate level of any of the most common engineering programmes.

Concerning the layout of the text; there are important keywords which will appear as *Margin text*. As a student reading the text for the first time one should, after reading a certain chapter or section, go back and use these margin texts as reminders for having reached a sufficient level of understanding of different important concepts.

Margin text

One major challenge when trying to describe Finite Elements is to give sufficient detail without making it too lengthy. That is the reason why some in-depth material is given at the end of the book rather than where it appears for the first time.

Notation

Notation principles used in this book are summarized below. If a letter or symbol is used twice with a different meaning, the letter or symbol will be given twice in this list.

As such, if a concept defined by a letter or symbol has several different used names describing the same and equivalent interpretation, all will be given below.

General mathematical symbols

a	A scalar value
$\boldsymbol{a}$	A column vector written as a **bold** lower-case letter
a_i	A coefficient in a vector
$\boldsymbol{A}$	A matrix written as a **bold** upper-case letter
$\mathcal{C}^0$	The set of all continuous functions
$\mathcal{C}^1$	The set of all functions having a continuous first-order derivative
$\mathbb{L}$	Local equations
$\mathbb{S}$	Strong formulation
$\mathbb{W}$	Weak formulation
$\mathbb{G}$	Galerkin formulation
$\mathbb{M}$	Matrix problem
n_{sd}	Number of spatial dimensions
n_{el}	Number of elements belonging to the mesh
n_n	Number of nodes belonging to the mesh
n_n^e	Number of nodes belonging to an element
n_f	Number of unknown freedoms in the mesh
n_p	Number of prescribed freedoms in the mesh

Latin symbols

A	An area or cross-sectional area
$\boldsymbol{a}$	The global unknown vector, the global displacement vector, the global degrees of freedom (d.o.f.) vector
$\boldsymbol{a}^e$	An element-local unknown vector, element-local degrees of freedom (d.o.f.) vector
$\boldsymbol{b}, b$	Load per unit volume or length
$\boldsymbol{B}$	Global kinematic matrix
$\boldsymbol{B}^e$	Element-local kinematic matrix
$\boldsymbol{C}^e$	Boolean connectivity matrix
$\boldsymbol{c}, c_i$	An arbitrary vector used for the weight function
$\boldsymbol{c}$	Element nodal coordinate vector
c_p	Specific heat coefficient
x_i, y_i, z_i	Nodal coordinate components
$\boldsymbol{D}$	Elasticity matrix in flexibility form
$\boldsymbol{D}$	Thermal conductivity matrix
E	Young's modulus of elasticity
$\boldsymbol{E}$	Elasticity matrix in stiffness form
$\boldsymbol{f}$	Global load vector
$\boldsymbol{f}_d$	Global load vector from internal distributed forces
$\boldsymbol{f}_g$	Global load vector from essential boundary conditions
$\boldsymbol{f}_h$	Global load vector from natural boundary conditions
$\boldsymbol{f}_r$	Global reaction force vector
G	Shear modulus
$\boldsymbol{G}$	Global operator matrix, gradient matrix
$\boldsymbol{g}, g$	Essential boundary conditions (multi-dim or 1D)
$\boldsymbol{h}, h$	Natural boundary conditions (multi-dim or 1D)
I	Area moment of inertia
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{K}$	Global stiffness or conductivity matrix
$\boldsymbol{K}^e$	Element stiffness or conductivity matrix
L	Length
M	Bending moment
$\boldsymbol{N}$	Global shape function matrix

N_i	A global shape function
$\boldsymbol{N}^e$	Element-local shape function matrix
N_i^e	A element-local shape function
N	Axial force in bars or beams
Q	Heat generation per unit length
$\boldsymbol{q}, q$	Heat flux per unit surface (multi-dim or 1D)
q_n	Heat flux perpendicular to the surface
q	Load per unit length
$\boldsymbol{R}$	Residual vector
$\boldsymbol{r}$	Unbalanced force vector or discrete residual vector
T	Temperature
T_∞	Surrounding temperature
T	Shear force
t	Time
t	Thickness
$\boldsymbol{t}$	Traction vector
t_i	A test function
S	Surface
S	Statical moment, the first moment
$\boldsymbol{S}$	Stress tensor
S_g	The part of the surface where essential boundary conditions are known
S_h	The part of the surface where natural boundary conditions are known
$\boldsymbol{s}$	Stress vector
$\boldsymbol{u}$	Displacement vector
u, v, w	Displacement vector components
$\boldsymbol{w}, w$	Weight function (vector-valued or scalar-valued)
V	Volume
x, y, z	Global coordinates

Greek symbols

α	Thermal expansion coefficient
α	Thermal convection coefficient
δ_{ij}	Kronecker delta

ϕ_i	A weight function
ε	Normal strain
$\boldsymbol{\varepsilon}$	Strain component vector
θ_i	Nodal rotation component
λ	Heat conductivity
ν	Poisson's ratio
ϱ	Density
σ_{ij}	Normal stress component
τ_{ij}	Shear stress component
ξ, η, ζ	Local coordinates

Chapter 1

Prelude

Nowadays Finite Elements are the standard tools for doing simulations in a large variety of engineering disciplines. Finite Elements are no more a tool for just a limited number of enthusiastic experts; they are something all of us as engineers have to learn.

One reason for why this method still, to some extent, is looked upon as a technique which you as an engineer can decide **not** to learn is probably because it is believed to be too difficult and time-consuming.

It is now time to change this option once for all. All of us can learn Finite Elements. Every engineer must know at least some basic facts from Finite Elements applied to some of the most important fields of application. When trying to learn Finite elements it is important and useful to have a solid knowledge of the physical problem, models of it and their analytical solutions. That is why Finite Elements should be studied in close connection to overall basic studies of a certain engineering discipline. Finite Elements is just a approximate numerical tool for solving some basic local equations constituting a mathematical model of reality.

A reason for why this technique is still looked upon as difficult to learn is probably that most text books are written by dedicated researcher in different fields of finite element applications. As an author one probably tends to describe the method from a mathematical point of view as consistently as possible and with a notation perhaps never previously seen by the students.

The technique is now so well-established that from a mathematical point of view most features and deficiencies are known in a variety different mathematical formulations of different important problems.

In this text we will learn why the method works, how the method works both analytically and numerically, how to use the method in typical daily engineering applications and, probably most important lesson what properties one should expect form the numerical approximation of the unknown entities. It is the author's intention to write a text covering details from the early beginning of a discussion of the model of reality, which we as engineers would like analyze, to the very end where we have the results from the finite element analysis.

General Features of the text:

- Every finite element application will start from the early beginning of its application with a discussion concerning which are the basic equations, why must they hold and what are the basic physical assumptions.
- Every important concept and expression will be deduced and the mathematical chain will be unbroken throughout the text.
- The mathematical language will be simple and concise
- The text will not be weighed down by any rigorous mathematical proof of important statements.
- Every finite element application will end up in one or more solved examples by the finite element program **TRINITAS**.
- The text will also serve as a theoretical description of what is implemented in this program

1.1 Background

Finite Elements have been described over the last decade in several different ways. In the early beginning it was described as a Rayleigh-Ritz method for elasticity problems and later on as a general tool for solving of partial differential equations of various kinds, always based on a so called weak formulation. From a mathematical point of view, the first description was based on Calculus of variations and a modern formulation is now based on Functional analysis and the theory of linear vector spaces.

Important basic work was done by Courant during the first part of the 1940's and the word "finite element" was coined in 1960 by Clough. Interest from engineers working with different aeronautical industrial applications was one of the main driving forces during the development of the finite element method. During the 1970's the first general-purpose commercial finite element packages were available and other engineering disciplines started to use the method. The development of different computer based support activities such as preprocessing of finite element input and postprocessing of finite element output, and the overall success of the method has become possible due the fast increasing computer power which has been going on in parallel. Today Finite Elements are one important cornerstone in the entire Computer-Aid Engineering (CAE) environment containing most engineering activities needed to be done in most engineering branches.

1.2 The Big Picture

Nowadays Finite Elements are used in a large variety of engineering disciplines. Typical fields are elasticity and heat transfer problems in solid bodies and acoustics and fluid flow problems in fluids. A large number of different linear or non-linear, steady state or transient problem classes exist. All these applications are sometimes called *Computational mechanics*. If the scope is even further extended, use of Finite Elements is also possible and straightforward in magnetic field and diffusion problems etc.. *Computational mechanics*

This text will concentrate on elasticity and heat transfer problems which are the most important applications of Finite Elements among all different computational mechanics disciplines.

A rather limited number of physical entities well-known by most mechanical engineers will be used in these formulations. In elasticity problems the displacement vector $\boldsymbol{u}$ and in heat transfer problems the temperature T is of great importance. In fluid flow problems the velocity vector $\boldsymbol{v}$, the pressure p and the density ρ are basic unknowns. In acoustic problems the pressure p once again is of great importance. Please observe that the velocity $\boldsymbol{v}$ is just the time derivative of the displacement $\boldsymbol{u}$. In several transient (that is time-dependent) problems we will also have need for further time derivatives such the acceleration vector $\boldsymbol{a}$.

In elasticity problems the stress components σ_{ij} and the strain components ε_{ij} will be important ingredients. In heat transfer problems we will also have to put focus on the heat flux vector $\boldsymbol{q}$. To be very detailed the list can be made longer but the general conclusion so far is that the total number of physical entities needed to be familiar with is rather limited even if we are discussing the entire field of computational mechanics.

Typical to models of all those disciplines is that they consist of a limited number of equations of different types.

The first group of equations to be brought up in this discussion is the *Balance law*s motivated from basic behaviour of nature. There is the *Newton's second law*, $\boldsymbol{f} = m\boldsymbol{a}$ requiring that all forces acting on a body most be in equilibrium. This balance law is the base for elasticity problems. In heat transfer problems the governing balance law is the *Conservation of Energy*, the first law in Thermodynamics. This equation only means that energy is undestroyable. There is also a third important balance law governing fluid flow problems; this is *Conservation of Mass*. These three balance laws govern most computational mechanics applications. In more complex, and probably non-linear applications, sometimes several or all of these balance laws have to be utilized. *Balance law*

A second group of equations is the *Constitutive relation*s. Typical to these equations are that they all are empirical equations established through experimental studies. Common to these equations are also that they try to describe the behavior of a solid material or a fluid in terms of some useful measures. In elasticity problems a first choice is the *generalized Hooke's law* and in heat transfer the *Fourier's law* is equally common. In fluid flow calculations a *New-* *Constitutive relation* *tonian fluid flow* behavior is the first and simplest choice for domain property characterization.

Scalar-valued

Vector-valued

Another important classification of a typical finite element formulation is whether the problem ends up in a *Scalar-valued* or *Vector-valued* problem. In the discussion to come we will find out that the displacement vector $\boldsymbol{u}$ will be the basic unknown and in the heat transfer problem the temperature T will be the basic unknown. That is the elasticity problem is a vector-valued problem and heat transfer problem is scalar-valued problem which be described in detail later on.

Compatibility relation

In cases where we are studying vector-valued problems, there is also a need for a relation coming from a third group of equations. The group referred to here is the group of *Compatibility relations*. Typical to this group of equations is that they try to predict how deformations in a matter will take place. Such equations will always put up some relations for how different components must related to each other. In scalar-valued problems there is never need for any relation belonging to this group.

System of linear algebraic equations

What has been discussed so far is what is typical or is in common between different mathematical formulations of different fields of application of the finite element method. Also, from a numerical point of view, several overall important comments can be made for what is typical or shared between different finite element applications. As a user of a finite element program it is probably equally important to be aware of what is going on in the computer during different types of finite element analysis. In elasticity and heat transfer steady-state problems we will find out what the computer has to solve of *System of linear algebraic equations*. In cases of studying time-dependent problems our mathematical discussion will end up in systems of algebraic coupled ordinary differential equations in time which have to be solved numerically by any of some time integration scheme. An important aspect of such time integration schemes is if the scheme is implicit or explicit. These schemes have different merits and where fields of application rarely overlaps.

Linear Eigenvalue problems

From a numerical point of view we will also find another typical group of *Linear Eigenvalue problems*. The most important applications are dynamic eigenvalue problems and linear buckling problems.

In non-linear problems one sooner or later has to introduce a linearization of the equations and from a numerical point of view an iterative scheme based on *Newton's method* has to be employed.

This section only tries to give the reader an overview of the topic. Perhaps some of the keywords discussed have been touched upon in some other courses or contexts. Some of the algorithms and numerical techniques needed here probably have been studied in previous mathematical courses.

If some of the material discussed here is hard to understand it is very natural because this is an overview and more detail will be given later on. This section will probably serve equally well as a summary and not only as an introduction of the topic.

Part I

Linear Static Elasticity

Chapter 2

Introduction

In most engineering activities where Solid Mechanics considerations have to be taken into account, a good start is to assume a linear structural response and a load that is applied in a quasi-static manner. This is one of the simplest models to study and such an analysis can be classified as a *Linear Static Elasticity* analysis. A huge majority of all engineering analysis work done, with the purpose of trying to investigate Solid Mechanics properties of a structure, belongs to this class of analysis and in many cases such an analysis will serve as a proper final result from which most overall engineering decisions can be taken.

Linear Static Elasticity

In this part of the text linear strain-displacement relations (small displacements) and linear elastic stress-strain displacements will be assumed. If also all boundary conditions are constant and independent of the applied load the structure will show a linear response. In this part of the text the discussion will also be limited to problems with quasi-static load application; no inertia forces will be included.

In Solid Mechanics there exists a sequence of approximation levels based on different displacement assumptions giving a true 3D deformable body more or less freedom to deform. In the following, several of the most important of these basic displacement assumption ideas will be discussed in terms of the basic local equations, strong and weak formulations, and finally appropriate finite element formulations. In the text below the discussion will start with the bar assumption which is the assumption that gives a real 3D body the least deforming possibilities.

In the following chapters there are also finite element formulations given for beams, 2D and 3D solids and finally, Mindlin-Reissner shell elements.

In all these chapters, motivated by different basic displacement assumptions, the entire chain of equations will be given. The experienced reader will quickly look through this and understand that very much of the structure and overall basic nature of the equations are closely related in-between these different formulations for bars, beams and solids. That is, basic relations could have been written more generally once and only referred to in the next chapters.

But the text to come is, as already mentioned, written with a goal that a chapter or an application should be self-contained with minimum requirement for jumping in one direction or another in the text. Another typical feature for the text is that every discussion will start at the early beginning of the application by a thorough discussion of the local equations constituting the model of reality.

After these element-specific topics the text will continue with general discussions concerning how to assemble and solve the system of linear algebraic equations. Different direct and iterative algorithms and techniques for finding the solution will be given.

In the last chapters in this Linear static elasticity part of the text are discussions of some further important aspects concerning how to treat and analyze linear static elasticity problems. Sometimes there is need for transformations of different kinds. For example, one probably would like to introduce a skew support not parallel to any of the global directions; that is there will be need for a transformation of one or several element stiffness matrices.

In a large typical industrial finite element analysis there is likely to be a need for combining different element types to each other. This can be done by the imposing of constraints on the system of equations. A large variety of different possibilities exist. The text will also cover how to numerically solve systems of equations containing constraints.

The last sections in the Linear static elasticity part of the text will actually discuss a problem which is non-linear. That is frictionless contact problems where the basic problem is to find the extent of the contact surface. The contact surface is the part of the boundary where two contacting deformable bodies only transmit compressive normal stresses. In a general case the extent of this surface is a result of the analysis and it has to be established by iterations. A force-displacement relation is in the general case non-linear because of change of contact surface. An obvious typical real situation is a ball or roller bearing.

Chapter 3

Bars

Consider a straight slender body with a smoothly changing cross section $A(x)$ and with a length L. Let us now assume that all loads applied to the body act in the direction of the extension of the body, which is the local horizontal x-direction, see figure 3.1. There is a distributed load, h per unit surface $[N/m^2]$ at the right end and a distributed load, $b(x)$ per unit length $[N/m]$, acting in the interior of the body. That is, the body will not be exposed to any bending loads and the body will only be stretched in it's own direction. If it is necessary to include bending of such a slender structure we have to move to the beam displacement assumption discussed in the next section.

In figure 3.1 the left end of the bar has been given a known prescribed displacement g, where $g << L$, and $E(x)$ is the Young's modulus of the material.

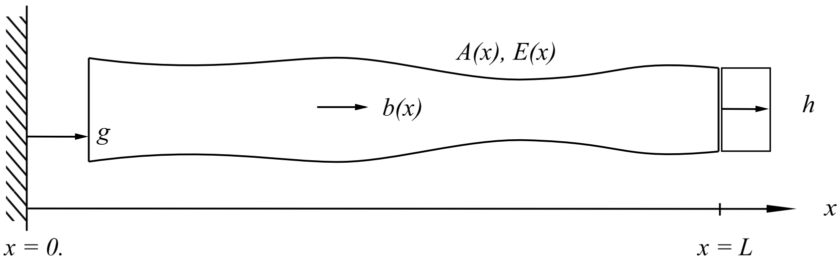


Figure 3.1: A typical bar structure

3.1 The Bar Displacement Assumption

Under the circumstances described above the *Bar displacement assumption* is *Bar displacement assumption* applicable. That is, every plane perpendicular to the x-axis is assumed to undergo just a constant translation in the x-direction and the initial plane will remain flat in it's deformed configuration.

By introducing this assumption the displacement u will be a function of x only as illustrated in figure 3.2.

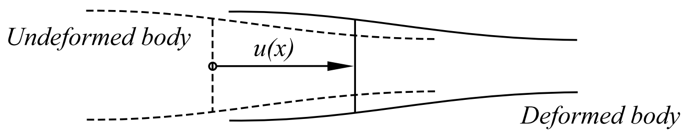


Figure 3.2: A typical bar deformation

This means that only one stress component $\sigma(x)$ and one strain component $\varepsilon(x)$ will be non-zero at every cut x through the bar. That is, from a mathematical point of view this problem is locally one-dimensional.

This model of reality will only involve three different unknown functions, the displacement $u(x)$, the strain $\varepsilon(x)$ and the stress $\sigma(x)$ in the interior of the body which has to be calculated under consideration of influence from the boundary conditions g and h.

3.2 The Local Equations

To be able to analyze this model there is need for at least three different equations.

As mentioned in the introduction, all models proposed for studying different physical phenomena always have to fulfill at least one balance law. In this case a statical equilibrium relation will serve as the balance law. Equilibrium for a

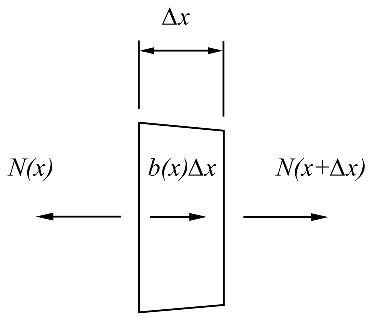


Figure 3.3: Forces acting on a slice Δx of a 1D bar model

short slice of length Δx of the bar requires

$$N(x + \Delta x) - N(x) + b(x)\Delta x = 0 \tag{3.1}$$

where Taylor's formula gives

$$N(x+\Delta x) \approx N(x) + \frac{dN(x)}{dx}\Delta x \tag{3.2}$$

and the axial force $N(x)$ can be expressed in the stress $\sigma(x)$ and the cross-sectional area $A(x)$ as follows

$$N(x) = A(x)\sigma(x). \tag{3.3}$$

These three equations (3.1) to (3.3) defines a *Balance Law* in terms of the stress $\sigma(x)$ and after division by Δx we have

Balance Law

$$\frac{d}{dx}\left(A(x)\sigma(x)\right) + b(x) = 0. \tag{3.4}$$

Typically, this equation always must hold independent from what stress-strain or strain-displacement relations will be assumed later on.

In this context, as we already have indicated, a linear elastic *Constitutive Relation* (the 1D Hooke's law)

Constitutive Relation

$$\sigma(x) = E(x)\varepsilon(x) \tag{3.5}$$

will be used and a linear compatibility relation (small displacements) can be deduced by using the displacement u at the two positions x and $x+\Delta x$ in figure 3.4.

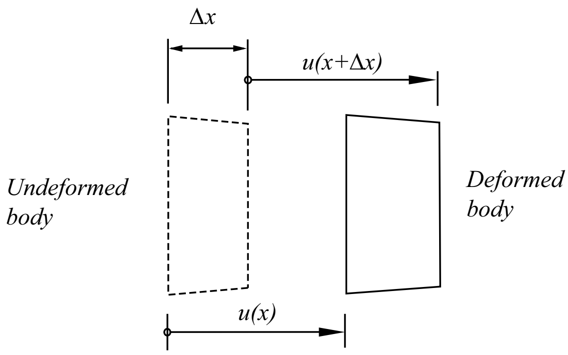


Figure 3.4: Typical bar deformation

The linear strain measure $\varepsilon(x)$ is defined as the change in length over the initial length Δx as follows

$$\varepsilon(x) = \frac{u(x+\Delta x) - u(x)}{\Delta x} = \frac{u(x) + \frac{du(x)}{dx}\Delta x - u(x)}{\Delta x} = \frac{du(x)}{dx} \tag{3.6}$$

Compatibility Relation

and this will serve as a *Compatibility Relation* for a linear 1D bar structure.

These three basic local equations 3.4 to 3.6 can be summarized in the box $\mathbb{L}$ as follows

Box: $\mathbb{L}$ ***'Local Equations in 1D Linear Static Elasticity'***

$$\frac{d}{dx}\left(A(x)\sigma(x)\right) + b(x) = 0$$

$$\sigma(x) = E(x)\varepsilon(x)$$

$$\varepsilon(x) = \frac{du(x)}{dx}$$

and these equations have to be fulfilled at any position inside the open domain $\Omega =]0; L[$. One obvious remark is of course that there is no influence form the boundary conditions so far.

3.3 A Strong Formulation

One of several possible ways to start the analytical work for solving this system of equations is to eliminate the stress $\sigma(x)$ and the strain $\varepsilon(x)$ by putting the constitutive relation and the compatibility relation into the balance law.

After introducing the boundary conditions from figure 3.1 the following well-posed boundary value problem $\mathbb{S}$ can be established.

Box: $\mathbb{S}$ ***'Strong form of 1D Linear Static Elasticity'***

Given $b(x)$, h and g. Find$u(x)$ such that

$$\frac{d}{dx}\left(A(x)E(x)\frac{du(x)}{dx}\right) + b(x) = 0 \qquad \forall \quad x \in \Omega$$

$$u(0) = g \qquad \text{on } S_g$$

$$E(L)\frac{du(L)}{dx} = h \qquad \text{on } S_h$$

Remarks:

1D second-order mixed Boundary-Value Problem

- This formulation $\mathbb{S}$ constitutes a Strong formulation of a linear static 1D Bar problem and from a mathematical point of view this is a *1D second-order mixed Boundary-Value Problem*

- $u(0) = g$ is a *non-homogeneous Essential* boundary condition. If $g = 0$ the boundary condition is homogeneous. *Essential*

- The total surface S consists in this 1D case only of the two end cross sections S_h and S_g.

- The differential equation is an example of a second order ordinary one.

- $E(L)du(L)/dx = h$ is a *Natural* boundary condition. *Natural*

- The boundary value problem is mixed because there are both essential and natural boundary conditions. Later on we will be aware of that some essential boundary conditions always have to exist to be able to guarantee the uniqueness of the solution of the matrix problem $\mathbb{M}$.

3.4 A Weak Formulation

A Strong formulation can always be transferred into an equivalent Weak formulation by multiplication of an arbitrary *Weight function* $w(x)$ and an integration over the domain. *Weight function*

$$\int_0^L w\left(\frac{d}{dx}(AE\frac{du}{dx}) + b\right) dx = \int_0^L w\frac{d}{dx}(AE\frac{du}{dx})\, dx + \int_0^L wb\, dx = 0 \quad (3.7)$$

After partial integration of the first term the following is obtained.

$$\left[wAE\frac{du}{dx}\right]_0^L - \int_0^L \frac{dw}{dx}AE\frac{du}{dx}\, dx + \int_0^L wb\, dx = 0 \quad (3.8)$$

The first term in the equation above can be rewritten as the natural boundary condition h can be identified form box $\mathbb{S}$ as

$$\left[wAE\frac{du}{dx}\right]_0^L = w(L)A(L)\underbrace{E(L)\frac{du(L)}{dx}}_{=h} - \underbrace{w(0)}_{=0}A(0)E(0)\frac{du(0)}{dx} \quad (3.9)$$

By putting one specific restriction on the weight function $w(x)$ and no longer letting the function $w(x)$ be completely arbitrary an infinite set $\mathcal{V}$ of functions can be defined where every choice of weight function $w(x)$ must be equal to zero on the part of the boundary where essential boundary conditions (S_g) are defined.

$$\mathcal{V} = \{w(x)|w(x) = 0 \quad \text{on } S_g\} \quad (3.10)$$

An appropriate Weak formulation $\mathbb{W}$ of this 1D Bar problem can be summarized as follows.

Box: $\mathbb{W}$ ***'Weak form of 1D Linear Static Elasticity'***

Given $b(x)$, h and g. Find$u(x)$ such that

$$\int_0^L \frac{dw(x)}{dx} A(x)E(x)\frac{du(x)}{dx}\,dx = \int_0^L w(x)b(x)\,dx + w(L)A(L)h$$

$$u(0) = g \qquad \text{on} \quad S_g$$

for all choices of weight functions $w(x)$ which belongs to the set V

Remarks:

- This weak formulation $\mathbb{W}$ serves as an efficient platform for applying numerical techniques such as weighted residual methods for solving this bar problem approximately.
- The partial integration step is performed because it opens the possibility to end up in a symmetric system of linear algebraic equations that is more efficiently solved in the computer compared to a non-symmetric system.
- The natural boundary condition is now implicitly contained in the integral equation.
- It is possible to show that the Strong and Weak formulations are equivalent.

3.5 A Galerkin Formulation

The basic reason for first turning the local equations into a Strong formulation and after that transfer the problem into an equivalent Weak formulation is that the weak form can be utilized as a base for a variety of different *Weighted Residual Methods* that all are capable of solving our basic bar problem, at least approximately.

Weighted Residual Methods

General to these methods are that both the unknown function $u(x)$ and the weight function $w(x)$ are built up from finite sums of n functions.

$$u(x) \approx u^h(x) = t_1(x)a_1 + t_2(x)a_2 + \ldots + t_n(x)a_n + t_0(x) = \sum_{i=1}^{n} t_i a_i + t_0(x) \quad (3.11)$$

and

$$w(x) = \phi_1(x)c_1 + \phi_2(x)c_2 + \ldots + \phi_n(x)c_n = \sum_{i=1}^{n} \phi_i c_i \quad (3.12)$$

Remarks:

- The functions $t_i(x)$ are called *Test functions*. Later on further details and rules will be given concerning how to select these functions and what properties they must fulfill. *Test functions*
- The function $t_0(x)$ must be there to secure that the non-homogeneous essential boundary condition $u(0) = g$ will be fulfilled. Further details will be given below.
- All a_i are unknown scalar constants. In the case when all test functions has been established the only unknowns are all these a_i.
- The arbitrariness of the weight function selection $w(x)$ is by this technique further limited to the choice of the n functions $\phi_i(x)$ and the value of each of the scalars constants c_i.
- By this introduction of finite series consisting of n functions our problem turns over from a *Continuous* one with infinite number unknowns to a *Discrete* one with a limited number of unknowns *Continuous* *Discrete*

One of the most popular weighted residual methods is the *Galerkin method*. One reason for this is that this method always will generate symmetric systems of linear algebraic equation which is more efficiently solved in the computer compared to non-symmetrical ones. Here the basic idea is *Galerkin method*

$$t_i(x) = \phi_i(x) = N_i(x) \qquad i = 1, 2, , , n \tag{3.13}$$

that if a selection is done of the test functions t_i every function ϕ_i also is defined and vice versa.

Please observe, that from here these functions (the test and the weight functions) most often will be called *Shape functions* and the notation $N_i(x)$ will be used. *Shape functions*

By moving over from sums to a matrix notation the approximation $u^h(x)$ and the weight function $w(x)$ can be rewritten as follow

$$u^h(x) = \boldsymbol{N}(x)\boldsymbol{a} + t_0 \tag{3.14}$$

$$w(x) = \boldsymbol{N}(x)\boldsymbol{c} \qquad \Rightarrow \qquad w(x) = \boldsymbol{c}^T \boldsymbol{N}^T(x) \tag{3.15}$$

where

$$\boldsymbol{N}(x) = \left[\, N_1(x) \;\; N_2(x) \;\; \ldots \;\; N_n(x) \,\right], \; \boldsymbol{a} = \left\{ \begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_n \end{array} \right\}, \; \boldsymbol{c} = \left\{ \begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_n \end{array} \right\}. \tag{3.16}$$

Concerning equation (3.15) the two alternatives are equal and actually the later will be mostly used.

Before some general and mathematically more precise rules will be given concerning what properties a certain choice of shape functions N_i have to fulfill, one possible choice among many others, will be given and discussed from an intuitive point of view.

In this particular bar problem we have now accepted an idea where an approximation is introduced for the displacement $u(x)$ in the bar. Later on we will find out that this will of course also generate approximate solutions for the stress and strain in the bar.

The simplest possible assumption is to think of the displacement approximation $u^h(x)$ as a piece-wise linear polygon chain. Such a function is continuous in its self but the first derivative is discontinuous. The question is now how to

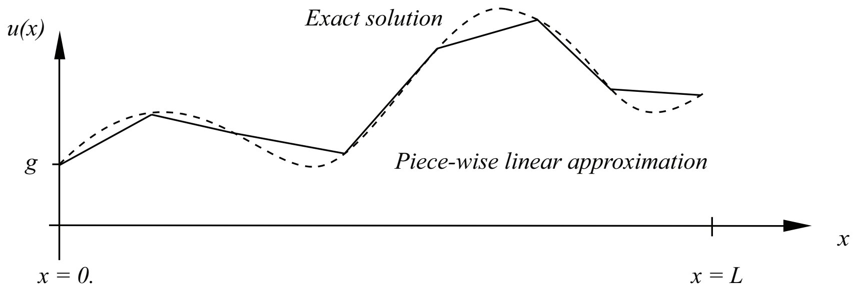


Figure 3.5: A 1D bar displacement approximation assumption

express such a piece-wise linear function as convenient and efficient as possible with n linear independent parameters typically stored in the column vector $\boldsymbol{a}$.

Nodes

Let a number of $n+1$ so called *Nodes* x_i be defined inside and at the ends of the domain Ω from 0 to L. The interval between two nodes is called a *Finite element*. As a first choice of functions N_i a set of piece-wise linear functions

Finite element

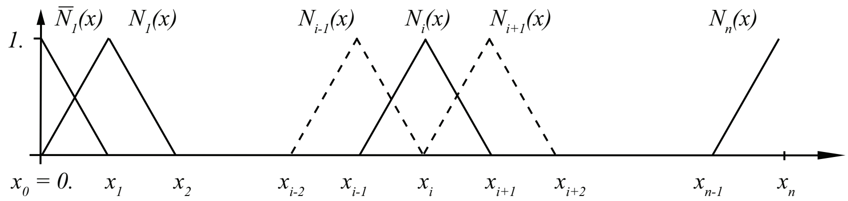


Figure 3.6: One possible choice of shape functions N_i for a 1D bar problem

will serve as a base for further discussions and so far they are only defined from intuitive reason and from figure 3.6 as follows

$$\bar{N}_1(x) = \begin{cases} (x_1 - x)/(x_1 - x_0) & 0. \leq x \leq x_1 \\ 0. & x_1 \leq x \leq x_n \end{cases} \tag{3.17}$$

$$N_i(x) = \begin{cases} 0. & x_0 \leq x \leq x_{i-1} \\ (x - x_{i-1})/(x_i - x_{i-1}) & x_{i-1} \leq x \leq x_i \\ (x_{i+1} - x)/(x_{i+1} - x_i) & x_i \leq x \leq x_{i+1} \\ 0. & x_{i+1} \leq x \leq x_n \end{cases} \tag{3.18}$$

$$N_n(x) = \begin{cases} 0. & x_0 \leq x \leq x_{n-1} \\ (x - x_{n-1})/(x_n - x_{n-1}) & x_{n-1} \leq x \leq x_n \end{cases} \tag{3.19}$$

Remarks:

- In a typical interval the unknown function will be approximated by a linear function as follows

  $$u(x) \approx u^h(x) = N_{i-1}(x)a_{i-1} + N_i(x)a_i \qquad \forall \qquad x_{i-1} \leq x \leq x_i \tag{3.20}$$

  where only two shape functions at the time will be non-zero and influence the approximation at an arbitrary point inside the interval.

- All these functions N_i have a unit value at one node and are zero at all other nodes. That is, the following holds

  $$N_i(x_j) = \delta_{ij} = \begin{cases} 1. & i = j \\ 0. & i \neq j \end{cases} \tag{3.21}$$

  which means that the shape functions are linear independent at the nodes.

- That is, the vector $\boldsymbol{a}$ will represent the displacement in the nodes.

- It is also possible to show that these shape functions $N_i(x)$ are linearly independent at an arbitrary position inside the intervals.

  From a general mathematical point it is possible to show that such a linearly independent choice of shape functions $N_i(x)$ will span a n-dimensional subspace from which the approximation will be received.

- Another important property that has to be fulfilled by a certain choice of a shape function $N_i(x)$ is that the function must belong to the set C^0 which consists of all continuous function $N_i(x)$ which fulfills

  $$\int_\Omega \left(\frac{dN_i(x)}{dx} \right)^2 dx < \infty. \tag{3.22}$$

- The function $t_0(x)$ can now be constructed from the $\bar{N}_1(x)$ function as follows

$$t_0(x) = \bar{N}_1(x)g \quad \Rightarrow \quad t_0(x \geq x_1) = 0. \tag{3.23}$$

By putting the equations (3.14), (3.15) and (3.23) into the weak formulation $\mathbb{W}$ the following discrete Galerkin formulation will be achieved.

$$\int_0^L \frac{d}{dx}(\boldsymbol{c}^T\boldsymbol{N}^T)AE\frac{d}{dx}(\boldsymbol{N}\boldsymbol{a} + \bar{N}_1 g)\,dx = \int_0^L \boldsymbol{c}^T\boldsymbol{N}^T b\,dx + \boldsymbol{c}^T\boldsymbol{N}^T(L)A(L)h$$

The vector $\boldsymbol{c}^T$ can be brought out as follows

$$\boldsymbol{c}^T\left\{\int_0^L \frac{d}{dx}\boldsymbol{N}^T AE\frac{d}{dx}(\boldsymbol{N}\boldsymbol{a} + \bar{N}_1 g)\,dx - \int_0^L \boldsymbol{N}^T b\,dx - \boldsymbol{N}^T(L)A(L)h\right\} = 0$$

and a matrix $\boldsymbol{B}(x)$ can be defined as

$$\boldsymbol{B}(x) = \frac{d\boldsymbol{N}(x)}{dx} = \begin{bmatrix} \frac{dN_1(x)}{dx} & \frac{dN_2(x)}{dx} & \dots & \frac{dN_n(x)}{dx} \end{bmatrix} \tag{3.24}$$

which then can be inserted into the equation above and the following is obtained

$$\boldsymbol{c}^T\left\{\int_0^L \boldsymbol{B}^T AE\boldsymbol{B}dx\,\boldsymbol{a} \quad - \right.$$
$$\left.\left(\int_0^L \boldsymbol{N}^T b\,dx + \boldsymbol{N}^T(L)A(L)h - \int_0^L \boldsymbol{B}^T AE\frac{d}{dx}\bar{N}_1 g\,dx\right)\right\} = 0.$$

Global stiffness matrix $\boldsymbol{K}$

Global load vector $\boldsymbol{f}$

The *Global stiffness matrix* $\boldsymbol{K}$ and the *Global load vector* $\boldsymbol{f}$ can be identified from this equation as

$$\boldsymbol{K} = \int_0^L \boldsymbol{B}^T(x)A(x)E(x)\boldsymbol{B}(x)dx \tag{3.25}$$

$$\boldsymbol{f} = \int_0^L \boldsymbol{N}^T(x)b(x)\,dx + \boldsymbol{N}^T(L)A(L)h -$$
$$\int_0^L \boldsymbol{B}^T(x)A(x)E(x)\frac{d}{dx}\bar{N}_1(x)g\,dx. \tag{3.26}$$

where the matrix $\boldsymbol{K}$ is a symmetric matrix with n rows and columns and the vector $\boldsymbol{f}$ is a column vector containing one load case.

A discrete Galerkin formulation for this 1D problem now reads

Box: $\mathbb{G}$ ***'Galerkin form of 1D Linear Static Elasticity'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{c}^T(\boldsymbol{K}\boldsymbol{a} - \boldsymbol{f}) = \boldsymbol{c}^T\boldsymbol{r} = 0$

for all choices of the vector $\boldsymbol{c}$ (the weight function)

Not necessary here, but convenient in the discussions to come is to introduce the following general split of the global load vector $\boldsymbol{f}$ into three different load vector contributions.

$$\boldsymbol{f} = \boldsymbol{f}_d + \boldsymbol{f}_h - \boldsymbol{f}_g \tag{3.27}$$

The first part $\boldsymbol{f}_d$ comes from internal distributed forces and in this 1D case it is equal to

$$\boldsymbol{f}_d = \int_0^L \boldsymbol{N}^T(x)b(x)\,dx \tag{3.28}$$

and two other parts are from essential and natural boundary conditions on S_h and S_g.

$$\boldsymbol{f}_h = \boldsymbol{N}^T(L)A(L)h \tag{3.29}$$

$$\boldsymbol{f}_g = \int_0^L \boldsymbol{B}^T(x)A(x)E(x)\frac{d}{dx}\bar{N}_1(x)g\,dx. \tag{3.30}$$

The vector $\boldsymbol{f}_h$ can always be evaluated, independent from the explicit choice of shape functions, as follows

$$\boldsymbol{f}_h = \boldsymbol{N}^T(L)A(L)h = A(L)h\begin{Bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{Bmatrix} \tag{3.31}$$

and vector $\boldsymbol{f}_g$ is shape function dependent. In the case with linear shape functions, as discussed so far, and a constant cross section A and Young's modulus E, we have

$$\boldsymbol{f}_g = \int_0^L \boldsymbol{B}^T AE\frac{d}{dx}\bar{N}_1 g\,dx = \frac{AE}{L_1^e}g\begin{Bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}. \tag{3.32}$$

where L_1^e is the length of the first element. From these expressions it is easy to conclude that the product $A(L)h$ and the product AEg/L_1^e are both forces.

3.6 A Matrix Formulation

It is obvious from above that the Galerkin formulation means a scalar product between the column vector $\boldsymbol{c}$ and another column vector $\boldsymbol{r}$ and it is still one single equation. The vector $\boldsymbol{r}$ is called the residual and it can be interpreted as *Unbalanced residual forces*.

Unbalanced residual forces

From the basic idea of involving an arbitrary weight function $w(x)$ in the weak formulation now only remains a vector $\boldsymbol{c}$. This vector still must be possible to select completely arbitrary. From this requirement it is obvious that the vector $\boldsymbol{r}$ must be equal to a zero vector

$$\boldsymbol{r} = \boldsymbol{0} \tag{3.33}$$

which means that the structure is in equilibrium. Please observe that this fulfillment of equilibrium is here said to be in a weak sense which means that we have equilibrium measured in nodal forces!

A matrix problem consisting of n linear algebraic equations can now be identified.

Box: $\mathbb{M}$ ***'Matrix form of 1D Linear Static Elasticity'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$

where $\boldsymbol{K}$ and $\boldsymbol{f}$ are known quantities

By solving this system of equations the vector $\boldsymbol{a}$ will represent the displacements at the nodes at equilibrium.

The very last step in the analysis is to calculate the, in the strong formulation eliminated stresses and strains, by making use of the compatibility and constitutive relations from the local equations (See box $\mathbb{L}$).

$$\varepsilon(x) = \frac{d}{dx}(\boldsymbol{N}(x)\boldsymbol{a} + \bar{N}_1(x)g) = \boldsymbol{B}(x)\boldsymbol{a} + \frac{d\bar{N}_1(x)}{dx}g \tag{3.34}$$

$$\sigma(x) = E(x)\frac{d}{dx}(\boldsymbol{N}(x)\boldsymbol{a} + \bar{N}_1(x)g) = E(x)(\boldsymbol{B}(x)\boldsymbol{a} + \frac{d\bar{N}_1(x)}{dx}g) \tag{3.35}$$

This is always done in an element-by-element fashion. What now lacks is a numerical procedure for establishing of the matrices $\boldsymbol{K}$ and $\boldsymbol{f}$ and solving of the matrix problem $\mathbb{M}$ for the vector $\boldsymbol{a}$. Such a numerical procedure is typically implemented as a computer program which can be called a *Finite Element Program*.

Finite Element Program

Remarks:

- In the beginning of this discussion there are three unknown functions of x. These are the stress $\sigma(x)$, the strain $\varepsilon(x)$ and the displacement $u(x)$ which all now can be calculated at least in an approximative manner.
- Most of the mathematical work done so far is of an analytical nature and needs only to be done once (when trying to learn and understand why the finite element method works before it comes to use of a computer program)
- Both the matrix $\boldsymbol{K}$ and vector $\boldsymbol{f}$ are completely defined by given data in figure 3.1, the number of shape functions N_i (elements) and the behavior of these shape functions (the element type)
- From a mathematical point of view this discussion can be summarized as

  $$\mathbb{L} \Rightarrow \mathbb{S} \Leftrightarrow \mathbb{W} \approx \mathbb{G} \Leftrightarrow \mathbb{M}$$

  and sources for errors in this mathematical model of reality are deviations from reality in the constitutive and the compatibility relations, deviations in the selected boundary conditions and numerical errors due to use of a limited number of Finite Elements with a specific behavior in each element.
- One can show that the solution to the matrix problem $\mathbb{M}$ always exists and has a unique solution if the global stiffness matrix $\boldsymbol{K}$ is non-singular. If there exist at least one essential boundary condition which prevents rigid body motion the stiffness matrix will be non-singular. That is the global stiffness matrix $\boldsymbol{K}$ is positive definite and the following holds

  $$\boldsymbol{a}^T \boldsymbol{K} \boldsymbol{a} > 0 \quad \forall \quad \boldsymbol{a} \neq \boldsymbol{0} \quad \Rightarrow \quad \det(\boldsymbol{K}) > 0$$
- In this 1D case the matrix $\boldsymbol{K}$ is symmetric with a three-diagonal population.

3.7 A 2-Node Element Stiffness Matrix

This 1D finite element analysis discussion is now approaching the end of the analytical part of the analysis and we are close to a position where we have to put in numbers and start the numerical part of the analysis. This is normally performed by a computer program based on this analytical discussion. What still has to be discussed is how to evaluate the integrals in box $\mathbb{M}$. After that the number and the behavior of shape functions N_i is decided, these integrals only contain known given quantities and the basic question is how to evaluated these as efficient as possible! Please observe, when it comes to practical use of a finite element program one always has to select a certain number of finite elements of a certain element type which means exactly the same as selecting the number and the behavior of the shape functions N_i.

One of the cornerstones in a finite element formulation is that the entire domain is split into a finite number of sub-domains, so called finite elements. Due to the nature of the shape functions as linear independent and only non-zero over very limited parts of the entire domain it is convenient to perform the integral over one element (one sub-domain) at a time and we have

$$\boldsymbol{K} = \sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \boldsymbol{B}^T(x)A(x)E(x)\boldsymbol{B}(x)dx \tag{3.36}$$

where this is a sum of n_{el} matrices where n_{el} is the number of finite elements. Each of these sub-matrices will only contain 4 non-zero coefficients symmetrically positioned around the main-diagonal of the sub-matrix. This is because the vector $\boldsymbol{B}$ evaluated for x-values inside the interval x_i and x_{i+1} will only contain 2 non-zero positions and the non-zero part of the product $\boldsymbol{B}^T\boldsymbol{B}$ is a symmetric 2 row and 2 column matrix.

Let us now study such a sub-interval in more detail. We will now move over to an element-local notation, see figure 3.7. The two linear parts of the global

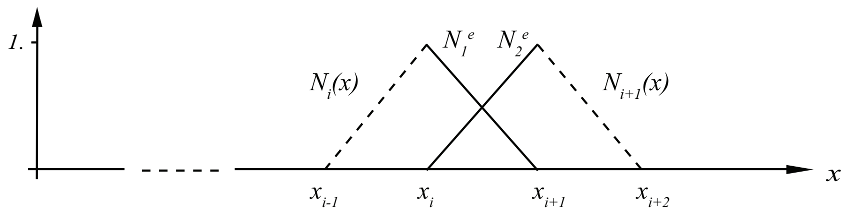


Figure 3.7: The relation between element-local and global shape functions

shape function N_i and N_{i+1} over the interval from x_i to x_{i+1} has been given the closely related notations N_1^e and N_2^e which is an element-local numbering from 1 to 2 over the number of nodes associated with this element. The following new element-local vectors can now be defined

$$\boldsymbol{N}^e(x) = \left[\begin{array}{cc} N_1^e(x) & N_2^e(x) \end{array}\right] \qquad \text{and} \qquad \boldsymbol{a}^e = \left\{\begin{array}{c} a_1^e \\ a_2^e \end{array}\right\} \tag{3.37}$$

and be used for an element-local expression of the displacement approximation $u^e(x)$ as follows

$$u^e(x) = N_1^e(x)a_1^e + N_2^e(x)a_2^e = \boldsymbol{N}^e(x)\boldsymbol{a}^e. \tag{3.38}$$

The element-local strain approximation $\varepsilon^e(x)$ can now be written as

$$\varepsilon^e(x) = \frac{du^e(x)}{dx} = \underbrace{\left[\begin{array}{cc} \dfrac{dN_1^e(x)}{dx} & \dfrac{dN_2^e(x)}{dx} \end{array}\right]}_{=\boldsymbol{B}^e(x)} \boldsymbol{a}^e = \boldsymbol{B}^e(x)\boldsymbol{a}^e \tag{3.39}$$

and the global matrix $\boldsymbol{B}$ will here appear in an element-local version $\boldsymbol{B}^e$.

A relation between the global unknown vector $\boldsymbol{a}$ and the *element-local* unknown vector $\boldsymbol{a}^e$ is easily established as

element-local

$$\boldsymbol{a}^e = \left\{ \begin{array}{c} a_1^e \\ a_2^e \end{array} \right\} = \underbrace{\left[\begin{array}{cccccc} 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \end{array} \right]}_{=\boldsymbol{C}^e} \left\{ \begin{array}{c} a_1 \\ \vdots \\ a_{i-1} \\ a_i \\ \vdots \\ a_n \end{array} \right\} = \boldsymbol{C}^e \boldsymbol{a} \qquad (3.40)$$

where the matrix $\boldsymbol{C}^e$ is a *Boolean Matrix* populated by only unity or zero values.

Boolean Matrix

Remarks:

- The element-local vector $\boldsymbol{a}^e$ is always a subset of the global vector $\boldsymbol{a}$.
- There is always one unity value in each row of the matrix $\boldsymbol{C}^e$ as long as none of the nodes in the element belongs to the boundary S_g, where we have known values of the displacements.
- In cases where one or several nodes are associated to the boundary S_g we can so far think of a matrix $\boldsymbol{C}^e$ preserving its number of rows and where a zero row without any unit value is introduced corresponding to the given value g.
- A more thorough and deepened discussion of this topic can be found in chapter 5 under section 5.8.

It can now be shown that the global stiffness matrix $\boldsymbol{K}$ can be built from a sum of small 2x2 matrices which are expanded by a pre- and post-multiplication of the boolean matrix $\boldsymbol{C}^e$.

$$\boldsymbol{K} = \sum_{i=1}^{n_{el}} \boldsymbol{C}_i^{e^T} \underbrace{\int_{x_i}^{x_{i+1}} \boldsymbol{B}^{e^T}(x) A(x) E(x) \boldsymbol{B}^e(x)\, dx}_{=\boldsymbol{K}_i^e} \; \boldsymbol{C}_i^e \qquad (3.41)$$

Such a small matrix is an important and often discussed topic called the *Element Stiffness Matrix* $\boldsymbol{K}^e$. The subscript i will only be used when a specific element i is discussed.

Element Stiffness Matrix

In this analytical discussion it is now time to perform the very last analytical steps. Let us express the two local shape functions N_1^e and N_2^e in an element-local coordinate axis ξ and in accordance to the figure 3.8 and we typically have

$$N_1^e(\xi) = \frac{1}{2}(1-\xi); \qquad N_2^e(\xi) = \frac{1}{2}(1+\xi). \qquad (3.42)$$

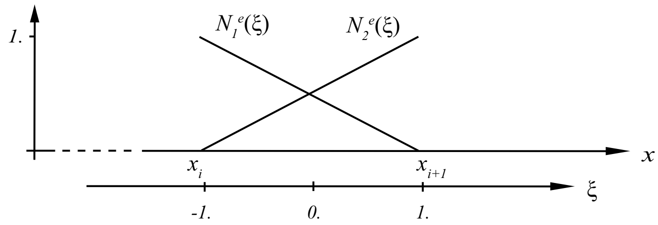


Figure 3.8: The element-local coordinate system ξ

The mapping between the two coordinate systems can be written as

$$x(\xi) = \frac{1}{2}(x_{i+1} - x_i)\xi + \frac{1}{2}(x_{i+1} + x_i). \tag{3.43}$$

where the length of the element $L^e = (x_{i+1} - x_i)$. Differentiation and the chain rule then gives

$$dx = \frac{1}{2}L^e d\xi \qquad \text{and} \qquad \frac{dN_i^e(\xi(x))}{dx} = \frac{N_i^e(\xi)}{d\xi}\frac{d\xi}{dx} = \frac{2}{L^e}\frac{N_i^e(\xi)}{d\xi} \tag{3.44}$$

and it is easy to evaluate the $\boldsymbol{B}^e$ matrix as follows

$$\boldsymbol{B}^e = \begin{bmatrix} \frac{dN_1^e}{dx} & \frac{dN_2^e}{dx} \end{bmatrix} = \frac{2}{L^e}\begin{bmatrix} \frac{dN_1^e}{d\xi} & \frac{dN_2^e}{d\xi} \end{bmatrix} = \frac{1}{L^e}\begin{bmatrix} -1 & 1 \end{bmatrix} \tag{3.45}$$

where the $\boldsymbol{B}^e$ matrix in this simple case is independent from the local coordinate system. The element stiffness matrix $\boldsymbol{K}^e$ is then

$$\boldsymbol{K}^e = \frac{1}{{L^e}^2}\begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix}\int_{-1}^{1} A(x)E(x)\frac{L^e}{2}\, d\xi \tag{3.46}$$

and if the the cross section $A(x)$ and the Young's modulus $E(x)$ are constants with respect to x and ξ we have

$$\boldsymbol{K}^e = \frac{EA}{L^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\frac{1}{2}\underbrace{\int_{-1}^{1} d\xi}_{=2} \tag{3.47}$$

which finally can be summarized in the box below.

Box: *'1D 2-node bar element stiffness matrix'*

$$\boldsymbol{K}^e = \frac{EA}{L^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

This is exactly what has to be implemented and evaluated numerically in the computer program and the end of the analytical discussion is reached at least for this element type. Even if the cross section $A(x)$ changes over the domain one typically use the value of the cross section at the mid-point of the element. That is, the the cross section is modeled as a step-wise constant function.

3.8 A 3-Node Element Stiffness Matrix

The selection of shape functions discussed so far is actually the simplest possible with its piece-wise linear nature with a discontinuous first-order derivative.

Let us now introduce a second choice of shape functions, still with a discontinuous first-order derivative, requiring a node at the mid-point of each element. By doing so our approximation of the displacement $u(x)$ will be enhanced by a second-order term and the approximation will be a piece-wise parabolic polynomial chain.

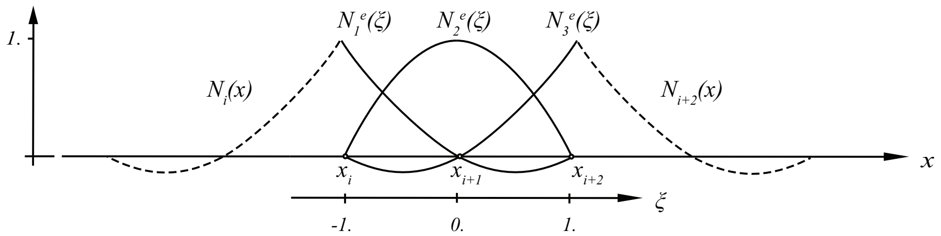


Figure 3.9: Element local shape functions for 3-node element

In a typical element of length $L^e = x_{i+2} - x_i$ we have now defined three element-local shape functions in accordance to figure 3.9 and element-local displacement approximation can be written as

$$u(\xi) = N_1^e(\xi)a_1^e + N_2^e(\xi)a_2^e + N_3^e(\xi)a_3^e = \boldsymbol{N}^e\boldsymbol{a}^e. \tag{3.48}$$

where

$$\boldsymbol{N}^e = [\ N_1^e(\xi) \quad N_2^e(\xi) \quad N_3^e(\xi)\ ] = \left[\ \frac{\xi}{2}(\xi-1) \quad 1-\xi^2 \quad \frac{\xi}{2}(\xi+1)\ \right]. \tag{3.49}$$

It is now possible to evaluate the $\boldsymbol{B}^e$ matrix as follows

$$\boldsymbol{B}^e = \left[\ \frac{dN_1^e}{dx} \quad \frac{dN_2^e}{dx} \quad \frac{dN_3^e}{dx}\ \right] = \frac{2}{L^e}\left[\ \frac{dN_1^e}{d\xi} \quad \frac{dN_2^e}{d\xi} \quad \frac{dN_3^e}{d\xi}\ \right] \tag{3.50}$$

where the $\boldsymbol{B}^e$ matrix in this parabolic case will be dependent on the local coordinate system. After introducing derivatives of the shape functions with respect to ξ we have

$$\boldsymbol{B}^e = \frac{2}{L^e} \left[\begin{array}{ccc} \xi - 1/2 & -2\xi & \xi + 1/2 \end{array} \right]. \tag{3.51}$$

The element stiffness matrix $\boldsymbol{K}^e$ will in this case be a 3x3 matrix and in a case with constant cross section and Young's modulus we have

$$\boldsymbol{K}^e = \frac{4AE}{L^{e^2}} \int_{-1}^{1} \left[\begin{array}{c} \xi - 1/2 \\ -2\xi \\ \xi + 1/2 \end{array} \right] \left[\begin{array}{ccc} \xi - 1/2 & -2\xi & \xi + 1/2 \end{array} \right] \frac{L^e}{2} \, d\xi \tag{3.52}$$

$\Rightarrow$

$$\boldsymbol{K}^e = \frac{2AE}{L^e} \int_{-1}^{1} \left[\begin{array}{ccc} (\xi - 1/2)^2 & -2\xi(\xi - 1/2) & (\xi - 1/2)(\xi + 1/2) \\ & 4\xi^2 & -2\xi(\xi + 1/2) \\ sym. & & (\xi + 1/2)^2 \end{array} \right] d\xi \tag{3.53}$$

By solving six different integral over polynomials in ξ we end up with an element stiffness matrix for a 1D 3-node element for second-order problems as defined in the box below.

Box: *'1D 3-node bar element stiffness matrix'*

$$\boldsymbol{K}^e = \frac{EA}{6L^e} \left[\begin{array}{ccc} 14 & 16 & -2 \\ 16 & 32 & 16 \\ -2 & 16 & 14 \end{array} \right]$$

3.9 An Element Load Vector

Element Load Vector

According to the global load vector $\boldsymbol{f}$, it is also possible to identify such a typical element contribution called the *Element Load Vector* $\boldsymbol{f}^e$ which can be evaluated over one element interval at a time.

$$\boldsymbol{f} = \sum_{i=1}^{n} \boldsymbol{C}_i^{e^T} \underbrace{\int_{x_i}^{x_{i+1}} \boldsymbol{N}^{e^T} b(x) \, dx}_{= \boldsymbol{f}_i^e} + \boldsymbol{f}_h - \boldsymbol{f}_g \tag{3.54}$$

In the general case, we will find later on that element nodal loads generated from different distributed load contributions can be integrated over the element domain. The global load vector contributions $\boldsymbol{f}_h$ and $\boldsymbol{f}_g$ will be discussed further in the next section.

Concerning the element load vector calculations from different types of distributed loads one typically restrict the variation inside the element to the variation used for the displacement approximation. That is, in our 1D case and focusing on the 2-node element with the distributed load/unit length $b(x)$, we have

$$b(x) = \boldsymbol{N}^e \boldsymbol{b}^e = \begin{bmatrix} N_1^e & N_2^e \end{bmatrix} \left\{ \begin{array}{c} b_1^e \\ b_2^e \end{array} \right\} = \frac{1}{2}(1-\xi) b_1^e + \frac{1}{2}(1+\xi) b_2^e \tag{3.55}$$

where the required input to the element are the intensity of the distributed load at the two nodes, b_1^e and b_2^e. This treatment leads to what is called a *Consistent Load Vector* . This means that the load must not change more rapidly than the displacement.

Consistent Load Vector

Also the element load vector $\boldsymbol{f}^e$ is most conveniently evaluated in a local coordinate system ξ. Thus

$$\boldsymbol{f}^e = \int_{-1}^{1} \begin{bmatrix} N_1^e(\xi) \\ N_2^e(\xi) \end{bmatrix} \begin{bmatrix} N_1^e(\xi) & N_2^e(\xi) \end{bmatrix} \frac{L^e}{2} d\xi \left\{ \begin{array}{c} b_1^e \\ b_2^e \end{array} \right\} \tag{3.56}$$

which means integration of each coefficient in a symmetric 2x2 matrix as follows

$$\boldsymbol{f}^e = \frac{L^e}{2} \begin{bmatrix} \int_{-1}^{1} N_1^{e^2}(\xi) d\xi & \int_{-1}^{1} N_1^e(\xi) N_2^e(\xi) d\xi \\ sym. & \int_{-1}^{1} N_2^{e^2}(\xi) d\xi \end{bmatrix} \left\{ \begin{array}{c} b_1^e \\ b_2^e \end{array} \right\}. \tag{3.57}$$

These three integrals are simple polynomials to integrate over a symmetric interval and we receive the following numerical values

$$\int_{-1}^{1} N_1^{e^2}(\xi) d\xi = \int_{-1}^{1} N_2^{e^2}(\xi) d\xi = \frac{2}{3} \quad \text{and} \quad \int_{-1}^{1} N_1^e(\xi) N_2^e(\xi) d\xi = \frac{1}{3}. \tag{3.58}$$

By putting these values into the element load vector $\boldsymbol{f}^e$ expression, we obtain

$$\boldsymbol{f}^e = \frac{L^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \left\{ \begin{array}{c} b_1^e \\ b_2^e \end{array} \right\}. \tag{3.59}$$

Let us simplify this expression to a situation with a constant load intensity $b_1^e = b_2^e = b_0$, we get

$$\boldsymbol{f}^e = \frac{b_0 L^e}{2} \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} \tag{3.60}$$

which has the obvious physical interpretation that the total force generated by the load intensity b_0 times the element length L^e is split into two equal concentrated forces acting at the ends of the element.

3.10 The Assembly Operation

Now all analytical details are discussed and we know what is needed for doing an implementation of such a finite element algorithm for analysis of 1D linear static elasticity problems in some programming language.

The overall matrix problem in box $\mathbb{M}$ with the global stiffness equations $\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$ can now be rewritten in terms of element stiffness matrices $\boldsymbol{K}_i^e$ and element load vector contributions $\boldsymbol{f}_i^e$ as follows

$$\sum_{i=1}^{n_{el}} \boldsymbol{C}_i^{e^T} \boldsymbol{K}_i^e \boldsymbol{C}_i^e \, \boldsymbol{a} = \sum_{i=1}^{n_{el}} \boldsymbol{C}_i^{e^T} \boldsymbol{f}_i^e + \boldsymbol{f}_h - \boldsymbol{f}_g \tag{3.61}$$

and boundary condition terms $\boldsymbol{f}_h$ and $\boldsymbol{f}_g$. After calculation of each element contribution an expansion and adding of these element contributions to the global stiffness matrix $\boldsymbol{K}$ and the global load vector $\boldsymbol{f}$ is performed. This numerical process is called the *Assembly Operation* of the global stiffness equations.

Assembly Operation

Here we typically only calculate and store non-zero coefficients in the upper right part of the global stiffness matrix in the computer memory. This system of equations can be solved by several different solution techniques such as both direct or iterative algorithms. After such a solution procedure we have numerical values in the global freedom vector $\boldsymbol{a}$ and the very last step in the analysis is to calculate the strains and the stresses.

3.11 Stress and Strain Calculations

The strain and the stress calculation is, as already mentioned, performed on the element level. By making use of equations 3.34, 3.35 and 3.40 we have

$$\varepsilon^e = \boldsymbol{B}^e \boldsymbol{a}^e = \boldsymbol{B}^e \boldsymbol{C}^{\boldsymbol{e}} \boldsymbol{a} \qquad \text{and} \qquad \sigma^e = E\varepsilon^e. \tag{3.62}$$

In the cases with the 2-node linear element the strain and the stress approximation will be rough. Because the $\boldsymbol{B}^{\boldsymbol{e}}$ matrix is constant and independent of space the strain and the stress will receive a constant value in each element. That is, the strain and stress approximation will be piece-wise constant over the domain and the strain and stress field are discontinues over the element borders. This is general also in multi-dimensional linear elasticity problems.

From a more general point of view, one important and obvious question in this context is which points should be used to compute the stresses and the strains? The answer is that associated to a typical element (or shape function selection) there are always a limit number of well-defined points inside the element which gives the most accurate strain and stress approximation. Such points are called *superconvergent points* which for the 2-node element is at the center of the element and for the 3-node element there are two superconvergent points at $\xi = \pm 1/\sqrt{3}$.

Another important aspect possible to discuss already in this 1D context is that these jumps in the strain and the stress approximation can be used as a measure of the error in the numerical approximation. Such an error measure can be utilized in a so called *Adaptive finite element analysis* where an iterative procedure is used and the domain is remeshed with the error in the region form the previous calculation as a measure for what size the elements should have in that region of the domain.

Adaptive finite element analysis

3.12 Multi-dimensional Truss Frame Works

This finite element formulation of 1D elasticity problems discussed so far is probably not that important as a tool in the daily engineering work but it is very important as a first application of the finite element method for getting used to and familiar with **why** and **how** the method works.

This 1D formulation can easily be extended to multi-dimensional cases of frame works of bars often called *Truss elements*. Such a truss element does not transmit moment at the end points and the overall reaction force in such multi-dimensional element always acts in the direction of the element. A frame work model with its moment-free connections is a simple, fast and powerful tool for analysis of real frame works in a variety of different engineering structures such as buildings, bridges, cranes and mast structures. As an analyst one has to be careful not to introduce non-physical mechanisms because of the moment-free connections between the members in the model of the frame work which seldom is present in the real structure.

Truss elements

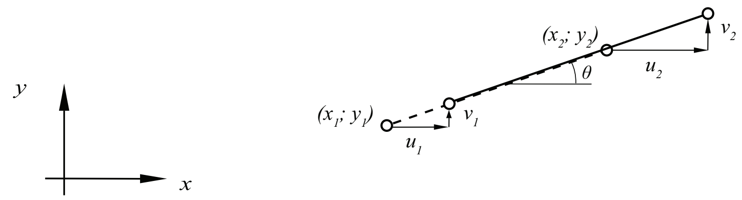


Figure 3.10: 1D bar element used in a 2D situation

Let us now think of our 1D coordinate system as a local direction in a global 2D or 3D coordinate system. The displacement of a 1D bar element expressed in a 2D global coordinate system can be written as follows

$$\boldsymbol{a}^e = \left\{ \begin{array}{c} a_1^e \\ a_2^e \end{array} \right\} = \left[\begin{array}{cccc} \Delta\bar{x} & \Delta\bar{y} & 0 & 0 \\ 0 & 0 & \Delta\bar{x} & \Delta\bar{y} \end{array} \right] \left\{ \begin{array}{c} u_1 \\ v_1 \\ u_2 \\ v_2 \end{array} \right\} = \boldsymbol{T}_{2D}\boldsymbol{a}_{2D}^e \tag{3.63}$$

or in a 3D case we receive

$$\boldsymbol{a}^e = \begin{bmatrix} \Delta\bar{x} & \Delta\bar{y} & \Delta\bar{z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta\bar{x} & \Delta\bar{y} & \Delta\bar{z} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \end{Bmatrix} = \boldsymbol{T}_{3D}\boldsymbol{a}^e_{3D} \tag{3.64}$$

where

$$\Delta\bar{x} = (x_2 - x_1)/L^e \quad \Delta\bar{y} = (y_2 - y_1)/L^e \quad \Delta\bar{z} = (z_2 - z_1)/L^e$$
$$L^e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

In the text to come, we will omit the subscripts $_{2D}$ and $_{3D}$ and rely on a context dependent notation where it is sufficient to drop these subscripts.

Please observe, the assumed displacement in figure 3.10 seems to be not very general but it is. It is the elongation of the bar that is important and as long as the rotation is small there are an infinite number of deformations ending up in the same elongation. The rotation doesn't influence the length of the bar in this linear small displacement analysis. In figure 3.11 an arbitrary deformed and rotated element is sketched with a scaled-up deformation.

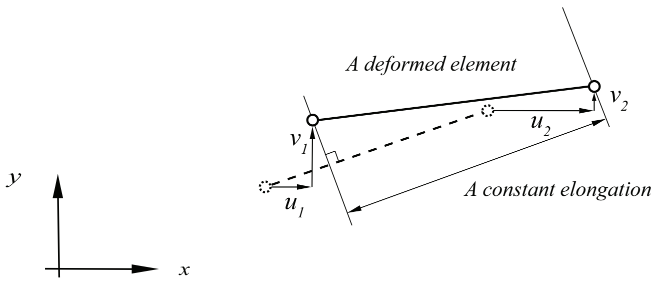


Figure 3.11: An arbitrary deformed element with a certain elongation

Let us pick up the equilibrium equation on the element level and for just one typical 1D element we have

$$\boldsymbol{K}^e\boldsymbol{a}^e = \boldsymbol{f}^e \quad \Leftrightarrow \quad \frac{EA}{L^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} a^e_1 \\ a^e_2 \end{Bmatrix} = \begin{Bmatrix} f^e_1 \\ f^e_2 \end{Bmatrix} \tag{3.65}$$

which always is two identical equations with opposite signs. The 1D element-local displacement vector a^e can be eliminated and in the 2D case we multiply with $\boldsymbol{T}^T_{2D}$ from the left and obtain

$$\underbrace{\boldsymbol{T}^T_{2D}\boldsymbol{K}^e\boldsymbol{T}_{2D}}_{=\boldsymbol{K}^e_{2D}}\boldsymbol{d}^e_{2D} = \boldsymbol{T}^T_{2D}\boldsymbol{f}^e \tag{3.66}$$

from which we can identify the element stiffness matrix for a 2D bar (or truss) element as follows

$$\boldsymbol{K}^e_{2D} = \frac{EA}{L^e}\begin{bmatrix} \Delta\bar{x} & 0 \\ \Delta\bar{y} & 0 \\ 0 & \Delta\bar{x} \\ 0 & \Delta\bar{y} \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \Delta\bar{x} & \Delta\bar{y} & 0 & 0 \\ 0 & 0 & \Delta\bar{x} & \Delta\bar{y} \end{bmatrix}. \tag{3.67}$$

After matrix multiplication we obtain

$$\boldsymbol{K}^e_{2D} = \frac{EA}{L^e}\begin{bmatrix} \Delta\bar{x}^2 & \Delta\bar{x}\Delta\bar{y} & -\Delta\bar{x}^2 & -\Delta\bar{x}\Delta\bar{y} \\ & \Delta\bar{y}^2 & -\Delta\bar{x}\Delta\bar{y} & -\Delta\bar{y}^2 \\ & & \Delta\bar{x}^2 & \Delta\bar{x}\Delta\bar{y} \\ sym & & & \Delta\bar{y}^2 \end{bmatrix}. \tag{3.68}$$

The 3D case is very similar as follows

$$\boldsymbol{K}^e_{3D} = \frac{EA}{L^e}\begin{bmatrix} \Delta\bar{x}^2 & \Delta\bar{x}\Delta\bar{y} & \Delta\bar{x}\Delta\bar{z} & -\Delta\bar{x}^2 & -\Delta\bar{x}\Delta\bar{y} & -\Delta\bar{x}\Delta\bar{z} \\ & \Delta\bar{y}^2 & \Delta\bar{y}\Delta\bar{z} & -\Delta\bar{x}\Delta\bar{y} & -\Delta\bar{y}^2 & -\Delta\bar{y}\Delta\bar{z} \\ & & \Delta\bar{z}^2 & -\Delta\bar{x}\Delta\bar{z} & -\Delta\bar{y}\Delta\bar{z} & -\Delta\bar{z}^2 \\ & & & \Delta\bar{x}^2 & \Delta\bar{x}\Delta\bar{y} & \Delta\bar{x}\Delta\bar{z} \\ & sym & & & \Delta\bar{y}^2 & \Delta\bar{y}\Delta\bar{z} \\ & & & & & \Delta\bar{z}^2 \end{bmatrix}. \tag{3.69}$$

These two element stiffness matrices are implemented as they are described above in most finite element packages such as in the **TRINITAS** program. When it comes to hand calculations for purpose of understanding the theory and the numerical work flow in a typical finite element analysis it is possible in 2D cases to identify an angle θ (See figure 3.10) where

$$\Delta\bar{x} = \cos\theta = c \qquad \Delta\bar{y} = \sin\theta = s$$

and the element stiffness matrix $\boldsymbol{K}^e_{2D}$ can be rewritten in a more "easy to use" form for hand calculations.

$$\boldsymbol{K}^e_{2D} = \frac{EA}{L^e}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ & s^2 & -cs & -s^2 \\ & & c^2 & cs \\ sym & & & s^2 \end{bmatrix} \tag{3.70}$$

3.13 Numerical Examples

A number of numerical examples will be studied below. Results from hand calculations and from computer calculations are presented.

A 1D bar problem

Consider an linear elastic bar of length L with a varying cross section $A(x)$

$$A(x) = A_0 \left(1 - \frac{x}{2L}\right)^2$$

and a given Young's modulus E. The bar is rigidly supported at the left end and at the right a concentrated force F is applied. The cross section variation is defined as follows

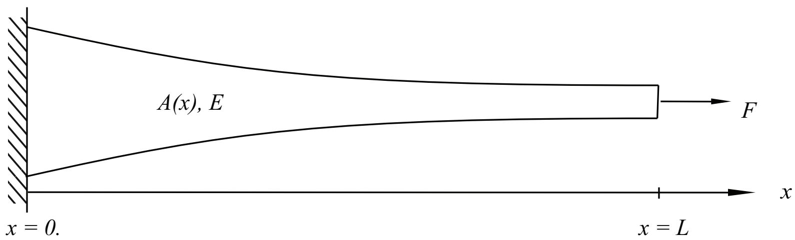


Figure 3.12: The given bar geometry

The exact solution for the displacement $u(x)$ can be achieved by solving of the strong formulation defined in box $\mathbb{S}$ where the given displacement g and the distributed force/unit length $b(x)$ are equal to zero. Putting in the cross section expression into the differential equation gives

$$\frac{d}{dx}\left(A_0 \left(1 - \frac{x}{2L}\right)^2 E \frac{du(x)}{dx}\right) = 0$$

and one integration of both sides $\Rightarrow$

$$EA_0 \left(1 - \frac{x}{2L}\right)^2 \frac{du(x)}{dx} = C_1$$

where C_1 is a unknown constant. In the right end we have the boundary condition

$$EA(L)\frac{du(L)}{dx} = A(L)h = F \quad \Rightarrow \quad C_1 = F.$$

A second integration then gives

$$u(x) = \frac{2FL}{EA_0} \frac{1}{\left(1 - \frac{x}{2L}\right)} + C_2$$

where the boundary condition $u(0) = 0 \Rightarrow$

$$C_2 = -\frac{2FL}{EA_0}$$

and the analytical solution for the displacement $u(x)$ is

$$u(x) = \frac{\frac{x}{2L}}{\left(1 - \frac{x}{2L}\right)} \frac{2FL}{EA_0}$$

and for the stress $\sigma(x)$ we have

$$\sigma(x) = E\frac{du(x)}{dx} = \frac{1}{\left(1 - \frac{x}{2L}\right)^2} \frac{F}{A_0}.$$

This expression will be used later on as a comparison.

Let us now solve this problem numerically by making use of the finite element method. Use the following finite element mesh consisting of three 2-node 1D

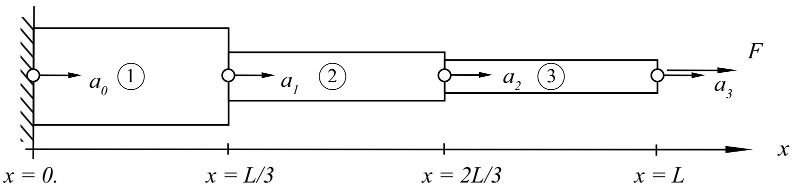


Figure 3.13: The given bar geometry

elements of equal length $L_i^e = L/3$ where $i = 1, 2, 3$. Concerning the cross section of the elements we here will use elements with a constant cross section calculated at the center of each element from the given expression. That is, $A_1 = A(x = L/6) = 121A_0/144$, $A_2 = A(x = L/2) = 81A_0/144$ and $A_3 = A(x = 5L/6) = 49A_0/144$ which gives the following element stiffness matrices

$$\boldsymbol{K}_1^e = \frac{121EA_0}{48L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \boldsymbol{K}_2^e = \frac{81EA_0}{48L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\boldsymbol{K}_3^e = \frac{49EA_0}{48L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Before assembling the global stiffness equations $\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$ we select a global node numbering in the unknown vector $\boldsymbol{a}$ as follows

$$\boldsymbol{a} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix}$$

and the Boolean matrices $\boldsymbol{C}_i^e$ can be identified as follows

$$\boldsymbol{C}_1^e = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{C}_2^e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{C}_3^e = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Rewriting of the global stiffness equation into a sum of expanded element stiffness matrices gives

$$\sum_{i=1}^{3} \boldsymbol{C}_i^{e^T} \boldsymbol{K}_i^e \boldsymbol{C}_i^e \boldsymbol{a} = \boldsymbol{f}$$

and $\Rightarrow$

$$\frac{EA_0}{48L} \left[\begin{bmatrix} 121 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 81 & -81 & 0 \\ -81 & 81 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 49 & -49 \\ 0 & -49 & 49 \end{bmatrix} \right] \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ F \end{Bmatrix}$$

which ends up in the following system to solve

$$\frac{EA_0}{48L} \begin{bmatrix} 202 & -81 & 0 \\ -81 & 130 & -49 \\ 0 & -49 & 49 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ F \end{Bmatrix}$$

where the solution is

$$\boldsymbol{a} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{160083} \frac{FL}{EA_0} \begin{Bmatrix} 63504 \\ 158368 \\ 315184 \end{Bmatrix}.$$

Finally, we can calculate strains and stresses in the elements from

$$\sigma_i^e = E\varepsilon_i^e = E\boldsymbol{B}^e \boldsymbol{a}_i^e = \frac{E}{L_i^e} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{Bmatrix} a_1^e \\ a_2^e \end{Bmatrix}_i$$

which gives

$$\sigma_1^e = \frac{3E}{L} \begin{bmatrix} -1 & 1 \end{bmatrix} \frac{FL}{160083EA_0} \begin{Bmatrix} 0 \\ 63504 \end{Bmatrix} = 63504F/53361A_0$$

$$\sigma_2^e = \frac{3E}{L} \begin{bmatrix} -1 & 1 \end{bmatrix} \frac{FL}{160083EA_0} \begin{Bmatrix} 63504 \\ 158368 \end{Bmatrix} = 94864F/53361A_0$$

$$\sigma_3^e = \frac{3E}{L} \begin{bmatrix} -1 & 1 \end{bmatrix} \frac{FL}{160083EA_0} \begin{Bmatrix} 158368 \\ 315184 \end{Bmatrix} = 158816F/53361A_0.$$

Please note that both the strain and the stress approximations inside this element type are constant. If we move over to 3-node element a piece-wise linear variation would be obtained.

Let us now put in the numerical values $L = 1.0\ m$, $F = 10000.\ N$, $E = 2.0{\cdot}10^{11}$ Pa and $A_0 = 0.01\ m^2$ which generates the following numerical results

$$a_1 \simeq 0.1983 \cdot 10^{-5}\ m \qquad a_2 \simeq 0.4946 \cdot 10^{-5}\ m \qquad a_3 \simeq 0.9844 \cdot 10^{-5}\ m$$

$$\sigma_1 \simeq 0.1190 \cdot 10^{7}\ Pa \qquad \sigma_2 \simeq 0.1778 \cdot 10^{7}\ Pa \qquad \sigma_3 \simeq 0.2939 \cdot 10^{7}\ Pa$$

and the analytical results are

$$u(L/3) = 0.2 \cdot 10^{-5}\ m \quad u(2L/3) = 0.5 \cdot 10^{-5}\ m \quad u(L) = 1.0 \cdot 10^{-5}\ m$$

$$\sigma(L/6) = 0.1190{\cdot}10^{7} Pa \quad \sigma(L/2) = 0.1778{\cdot}10^{7} Pa \quad \sigma(5L/6) = 0.2939{\cdot}10^{7} Pa$$

The same analysis has been done by the **TRINITAS** program and results are shown in figure 3.14 below. All freedom perpendicular to the bar have been fixed because there is no 1D bar element implemented in the program.

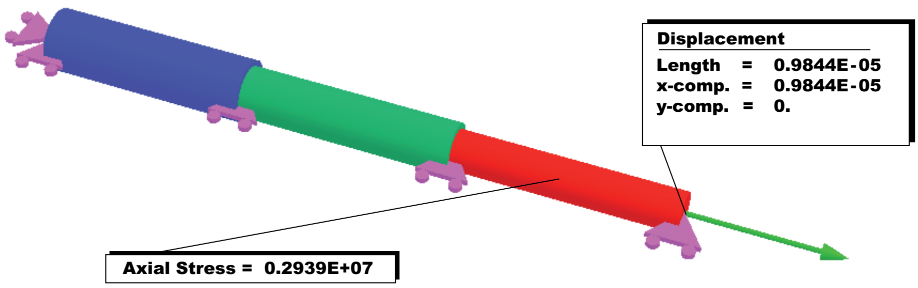


Figure 3.14: A **TRINITAS** analysis

Remarks:

- The approximation of the displacement always gives the best agreement at the nodes. In this case we have a relative error about one percent.

- A comparison of the stresses at the center of the element show an exact agreement. It can be shown that this is not just luck in this case and that this always holds for exactly this type problem. In the general case one should conclude that the stresses are less accurately approximated compared to the displacements.

- The stress field is in the general case always discontinuous at the element borders and no equilibrium equation in stress is fulfilled at element boundaries.

A 2D truss problem

In this example we will focus on a simple truss frame work consisting of three bar members with moment-free connections. This problem will be numerically analyzed by hand calculations in accordance to the finite element method. The

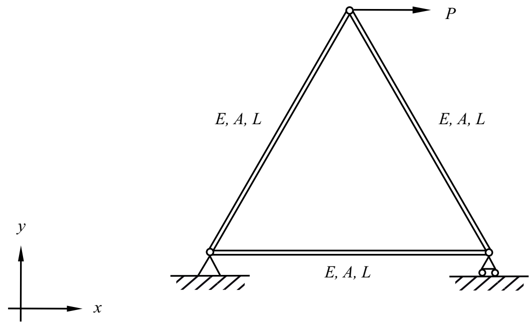


Figure 3.15: A 2D truss problem

frame work is modeled by three 2D bar elements and the global numbering of freedoms and elements are done in accordance to figure 3.16. The contents of

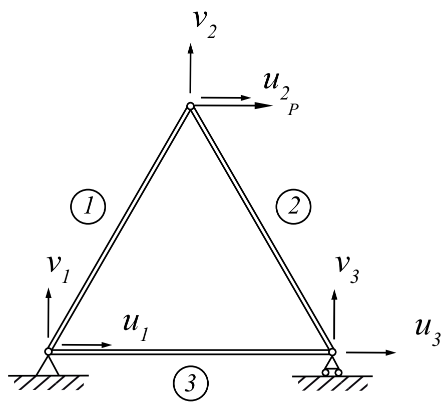


Figure 3.16: A 2D finite element mesh

the global freedom vector $\boldsymbol{a}$ can be established from this finite element mesh numbering. We have

$$\boldsymbol{a} = \left\{ \begin{array}{c} u_2 \\ v_2 \\ u_3 \end{array} \right\}$$

where the sequence in-between the unknowns is selected by the analyst and the selection will influence the appearance but not result of the entire analysis.

The element stiffness matrices $\boldsymbol{K}_i^e$ can be established from equation 3.70

$$\boldsymbol{K}_1^e = \frac{EA}{4L}\begin{bmatrix} 1 & \sqrt{3} & -1 & -\sqrt{3} \\ & 3 & -\sqrt{3} & -3 \\ & & 1 & \sqrt{3} \\ & & & 3 \end{bmatrix} \quad \boldsymbol{K}_2^e = \frac{EA}{4L}\begin{bmatrix} 1 & -\sqrt{3} & -1 & \sqrt{3} \\ & 3 & \sqrt{3} & -3 \\ & & 1 & -\sqrt{3} \\ & & & 3 \end{bmatrix}$$

$$\boldsymbol{K}_3^e = \frac{EA}{L}\begin{bmatrix} 1 & 0 & -1 & 0 \\ & 0 & 0 & 0 \\ & & 1 & 0 \\ & & & 0 \end{bmatrix}$$

and the Boolean matrices are

$$\boldsymbol{C}_1^e = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{C}_2^e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{C}_3^e = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

We are now ready to establish the global stiffness equation as a sum of expanded element stiffness matrices and we obtain as an intermediate result

$$\sum_{i=1}^{3} \boldsymbol{C}_i^{e^T} \boldsymbol{K}_i^e \boldsymbol{C}_i^e \boldsymbol{a} = \boldsymbol{f}$$

$\Rightarrow$

$$\frac{EA}{4L}\left[\begin{bmatrix} 1 & \sqrt{3} & 0 \\ \sqrt{3} & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -\sqrt{3} & -1 \\ -\sqrt{3} & 3 & \sqrt{3} \\ -1 & \sqrt{3} & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}\right]\begin{Bmatrix} u_2 \\ v_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \\ 0 \end{Bmatrix}$$

and the final system to solve then reads

$$\frac{EA}{4L}\begin{bmatrix} 2 & 0 & -1 \\ 0 & 6 & \sqrt{3} \\ -1 & \sqrt{3} & 5 \end{bmatrix}\begin{Bmatrix} u_2 \\ v_2 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \\ 0 \end{Bmatrix}$$

which has the following solution.

$$\boldsymbol{a} = \begin{Bmatrix} u_2 \\ v_2 \\ u_3 \end{Bmatrix} = \frac{FL}{4EA}\begin{Bmatrix} 9 \\ -1/\sqrt{3} \\ 2 \end{Bmatrix}$$

The axial stress in each of the bar elements can be calculated from

$$\sigma_i^e = E\boldsymbol{B}^e\boldsymbol{a}_i^e = E\boldsymbol{B}^e\boldsymbol{T}_i\boldsymbol{d}_i = \frac{E}{L_i^e}\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} \Delta\bar{x} & \Delta\bar{y} & 0 & 0 \\ 0 & 0 & \Delta\bar{x} & \Delta\bar{y} \end{bmatrix}_i \begin{Bmatrix} u_1^e \\ v_1^e \\ u_2^e \\ v_2^e \end{Bmatrix}_i$$

$\Rightarrow$

$$\sigma_i^e = \frac{E}{L_i^e} \left[\begin{array}{cccc} -\Delta\bar{x} & -\Delta\bar{y} & \Delta\bar{x} & \Delta\bar{y} \end{array} \right]_i \left\{ \begin{array}{c} u_1^e \\ v_1^e \\ u_2^e \\ v_2^e \end{array} \right\}_i$$

which gives for the three different elements used in this analysis.

$$\sigma_1^e = \frac{F}{8A} \left[\begin{array}{cccc} -1 & -\sqrt{3} & 1 & \sqrt{3} \end{array} \right] \left\{ \begin{array}{c} 0 \\ 0 \\ 9 \\ -1/\sqrt{3} \end{array} \right\} = \frac{F}{A}$$

$$\sigma_2^e = \frac{F}{8A} \left[\begin{array}{cccc} 1 & -\sqrt{3} & -1 & \sqrt{3} \end{array} \right] \left\{ \begin{array}{c} 2 \\ 0 \\ 9 \\ -1/\sqrt{3} \end{array} \right\} = -\frac{F}{A}$$

$$\sigma_3^e = \frac{F}{4A} \left[\begin{array}{cccc} -1 & 0 & 1 & 0 \end{array} \right] \left\{ \begin{array}{c} 0 \\ 0 \\ 2 \\ 0 \end{array} \right\} = \frac{F}{2A}$$

A numerical analysis of this problem has also been done by the finite element program **TRINITAS**. Used numerical values are

$$F = 10\,000.\ N, \quad L = 1.0\ m, \quad E = 2.0 \cdot 10^{11}\ Pa, \quad A = 0.001\ m^2$$

and the results are shown in figure 3.17

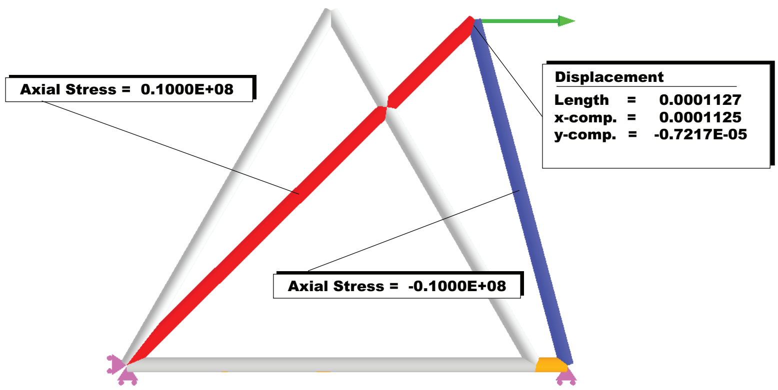


Figure 3.17: A **TRINITAS** 2D truss analysis

A 3D truss problem

Consider the following simple 3D truss frame work analyzed by the **TRINITAS** programs. The mesh consists of 4 nodes in 6 planes which means 24 nodes and $5(12 + 6) = 90$ 2-node 3D truss elements. All elements have a cross sectional area $A = .0005\, m^2$ and a Young's modulus $E = 0.2 \cdot 10^{11} Pa$. The applied load is a concentrated force vector $\boldsymbol{P} = \{10000. \quad 20000. \quad 0.\ \}^T\ N$. The geometrical positions of the nodes and some results are shown in figure 3.18. The geometry is defined by the distances $a_x = a_y = a_z = 0.4\, m$. All 4 nodes at the bottom are fully fixed in all directions.

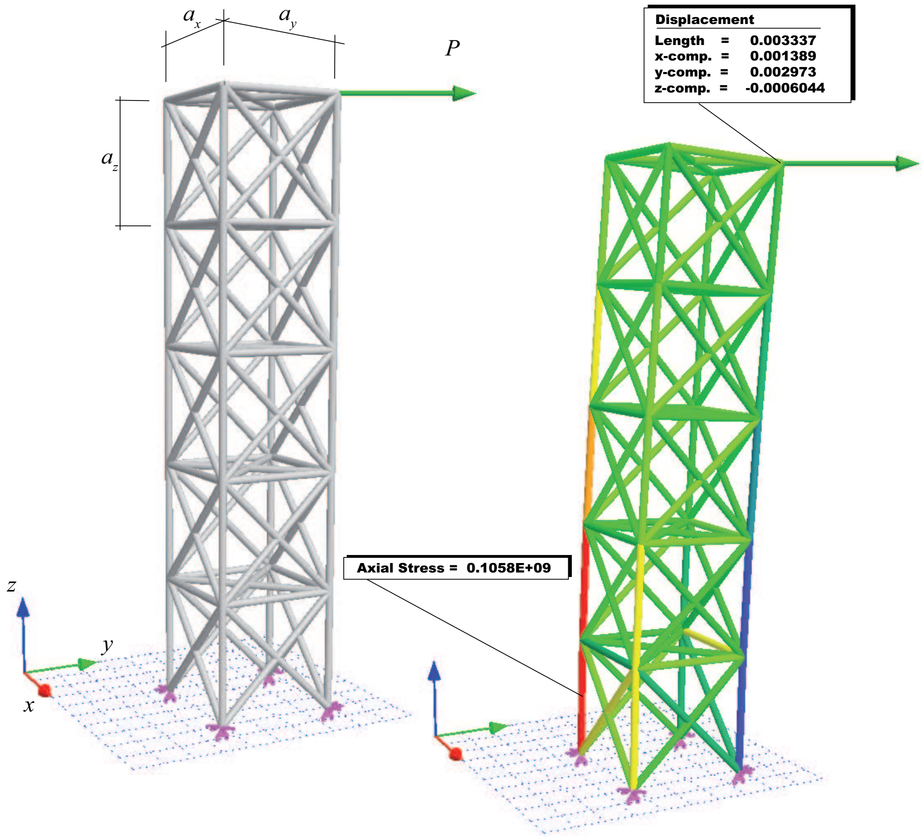


Figure 3.18: A **TRINITAS** 3D truss analysis

3.14 Common Pitfalls and Mistakes

As already mentioned rigid body motion always have to be prevented. That is at least one fixed or prescribed freedom in 1D problems, 3 fixed or prescribed freedoms in 2D cases or 6 fixed or prescribed freedoms in 3D cases have to be present. Please be very careful when it comes to suppressing of the rigid body rotations especially in 3D cases because it can actually be rather tricky. Besides these necessities to prevent rigid body motion due to the uniqueness of the solution a couple of common pitfalls will be discussed below.

Let us start the discussion by considering a mistake often made when applying symmetric boundary conditions. Figure 3.19 shows a symmetric 2D truss.

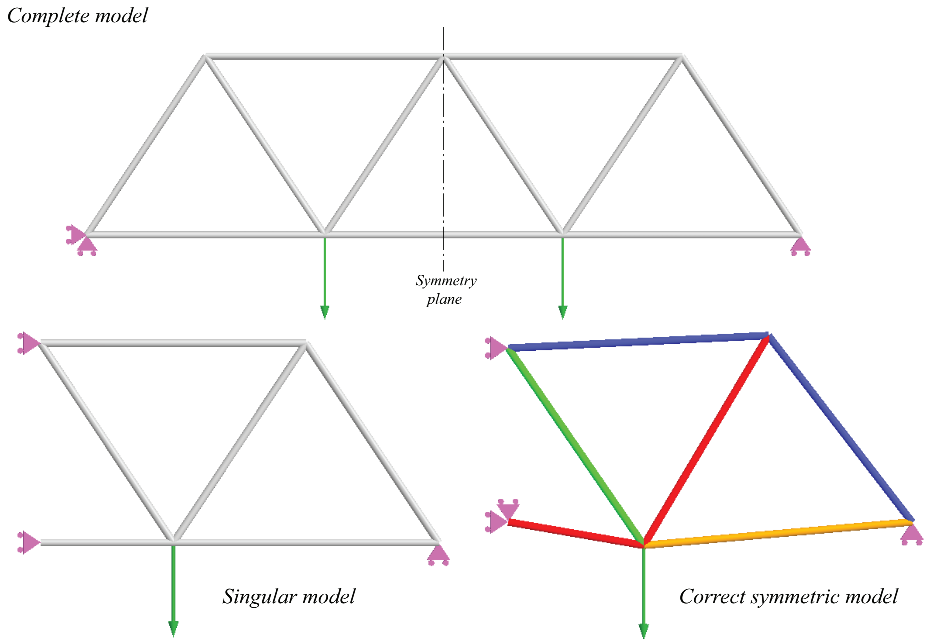


Figure 3.19: A 2D symmetric truss problem

The conclusion from this is that a single element can never be left without support in both ends. The lower-left element in the singular model above is free to rotate because the moment-free connection at the right end and the boundary condition at the left end which makes it possible to move freely in the vertical direction without generating any strain the element. From an intuitive point of view most engineers have a feeling that the element will stiffen because it will be longer but this effect is a non-linear effect approximated away and not taken into account in our linear compatibility relation.

Another closely related problem is mechanisms due to too few diagonal members in the structure. A simple 2D bridge-like structure will serve as an example.

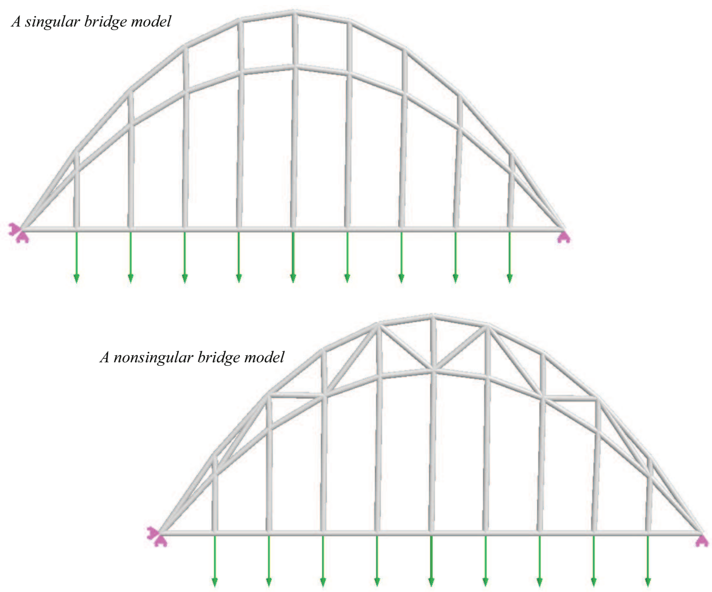


Figure 3.20: A 2D mechanism problem

The conclusion is that it is not necessary to put in diagonal members everywhere in the model. It is enough to prevent all overall possible mechanisms in the structure.

In 3D cases both those two types of problems discussed in the 2D problems above will be ever more frequent. As a very last example we discuss the the 3D case studied in the previous example. Think of a case where we also put in a node in the crossing point of each pair of diagonal members in the original structure. In this case each of these diagonal members will be split into two truss element and the model will be singular. The reason is that all these new extra nodes will not have any stiffness in a direction perpendicular to the plane spanned by the neighboring elements. This can easily be cured by just putting in a fixed boundary condition in the appropriate direction.

Generally one can conclude that use of truss frame work models, especially in 3D, requires a thorough understanding of what such a model with its moment-free connections means and how this property, which in most cases deviates from reality, effects the analysis. In many 3D cases it is often easier to create a beam model of the frame work but this is more time-consuming to solve compared to a bar model and it is often possible to draw the same overall engineering conclusions from the faster bar model.

3.15 Summary

In every plane in a 1D bar we have three unknowns and three local equations

Box: $\mathbb{L}$ ***'Local Equations in 1D Linear Static Elasticity'***

$$\frac{d}{dx}\left(A(x)\sigma(x)\right) + b(x) = 0$$

$$\sigma(x) = E(x)\varepsilon(x)$$

$$\varepsilon(x) = \frac{du(x)}{dx}$$

to fulfill. These equations can be turned over to a well-posed strong formulation by elimination of the strain and the stress and imposing of boundary conditions.

Box: $\mathbb{S}$ ***'Strong form of 1D Linear Static Elasticity'***

Given $b(x)$, h and g. Find$u(x)$ such that

$$\frac{d}{dx}\left(A(x)E(x)\frac{du(x)}{dx}\right) + b(x) = 0 \qquad \forall \quad x \in \Omega$$

$$u(0) = g \qquad \text{on } S_g$$

$$E(L)\frac{du(L)}{dx} = h \qquad \text{on } S_h$$

An equivalent weak form can be achieved by introduction of a weight function and after one partial integration we have

Box: $\mathbb{W}$ ***'Weak form of 1D Linear Static Elasticity'***

Given $b(x)$, h and g. Find$u(x)$ such that

$$\int_0^L \frac{dw(x)}{dx}A(x)E(x)\frac{du(x)}{dx}\,dx = \int_0^L w(x)b(x)\,dx + w(L)A(L)h$$

$$u(0) = g \qquad \text{on} \quad S_g$$

for all choices of weight functions $w(x)$ which belongs to the set $\mathcal{V}$

where the natural boundary conditions are implicitly contained. By introducing that the test functions and the weight functions are selected equally we end up in the following discrete Galerkin formulation.

Box: $\mathbb{G}$ ***'Galerkin form of 1D Linear Static Elasticity'***

Find $\boldsymbol{a}$ such that

$$\boldsymbol{c}^T(\boldsymbol{K}\boldsymbol{a} - \boldsymbol{f}) = \boldsymbol{c}^T\boldsymbol{r} = 0$$

for all choices of the vector $\boldsymbol{c}$(the weight function) where

$$\boldsymbol{K} = \int_0^L \boldsymbol{B}^T(x)A(x)E(x)\boldsymbol{B}(x)dx$$

$$\boldsymbol{f} = \int_0^L \boldsymbol{N}^T(x)b(x)\,dx + \boldsymbol{N}^T(L)A(L)h - \int_0^L \boldsymbol{B}^T(x)AE\frac{d\bar{N}_1(x)}{dx}g\,dx.$$

This is still only one single equation where now the essential boundary conditions are implicitly contained. Because of the arbitrariness of the vector $\boldsymbol{c}$ which implies that the residual $\boldsymbol{r}$ must be equal to zero the following system of linear algebraic equations is obtained.

Box: $\mathbb{M}$ ***'Matrix form of 1D Linear Static Elasticity'***

Find $\boldsymbol{a}$ such that

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$$

where $\boldsymbol{K}$ and $\boldsymbol{f}$ are known quantities

$$\boldsymbol{K} = \sum_{i=1}^{n_{el}} \boldsymbol{C}_i^{e^T}\boldsymbol{K}_i^e\boldsymbol{C}_i^e \qquad \boldsymbol{K}_i^e = \int_{x_i}^{x_{i+1}} \boldsymbol{B}^{e^T}(x)A(x)E(x)\boldsymbol{B}^e(x)\,dx$$

$$\boldsymbol{f} = \sum_{i=1}^{n_{el}} \boldsymbol{C}_i^{e^T}\boldsymbol{f}_i^e + \boldsymbol{f}_h - \boldsymbol{f}_g \qquad \boldsymbol{f}_i^e = \int_{x_i}^{x_{i+1}} \boldsymbol{N}^{e^T}(x)b(x)\,dx$$

The numerical procedure starts from here and the following work flow can be identified.

Numerical Work Flow:

- Split the entire domain into a number of Finite Elements of a certain type
- Define domain properties such as the Young's modulus E and the cross section A in the elements
- Define essential boundary conditions such as fixed or prescribed node displacements

- Define natural boundary condition as given concentrated or distributed loads
- Make a global numbering sequence of all involved unknown freedoms (Normally done automatically by the program)
- Calculate all element stiffness matrices $\boldsymbol{K}_i^e$ and expand and add those stiffness coefficients into the appropriate positions in the global stiffness matrix $\boldsymbol{K}$
- Calculate all element load vectors $\boldsymbol{f}_i^e$ and expand and add those load contributions into the appropriate positions in the global load vector $\boldsymbol{f}$
- Solve the system of linear algebraic equation $\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$ by some Gauss' elimination or LR-factorization look-a-like procedure. (Further discussions concerning how to calculate the unknown vector $\boldsymbol{a}$ will be given later on in the chapters to come)
- Pick up the element freedom vector $\boldsymbol{a}_i^e$ from the global one $\boldsymbol{a}$ for one element at the time and calculate the strains and the stresses
- Investigate the results, hopefully in terms of a nice color picture showing the deformed structure with the stress levels in color, and try to examine the relevance of the achieved approximation.

In most finite element analyses, after the results have been accepted from an overall engineering point of view, one also have to accept the analysis from a numerical point of view which normally means a refinement of the mesh in some critical part of the domain trying to find out if the accuracy of the numerical results is sufficient.

At this very end, one should not forget that the reason for doing the finite element analysis was some overall engineering question concerning how to design a certain piece of equipment and the finite element analysis only gives some hints concerning the size of displacements, strains and stresses in a **model of reality**.

Chapter 4

Beams

Let us now once again consider a straight slender body with a minor changing cross section $A(x)$ and area moment of inertia $I(x)$. Observe, that the geometry of this structure may be the same as the one we studied in the previous bar problem. But in this case, we no more restrict all loads to only act in the local x-direction. Here we will instead consider a distributed load $q(x)$ per unit length $[N/m]$ acting perpendicular to the local direction x and an important unknown is the displacement $v(x)$ in the local y-direction.

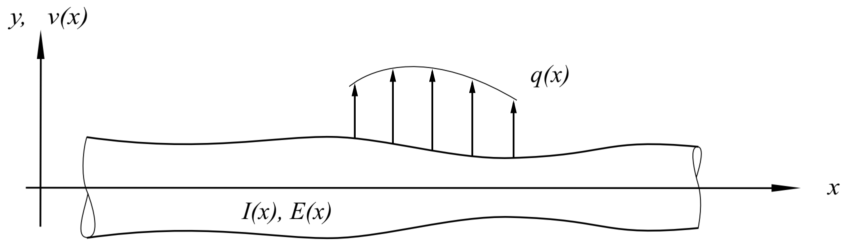


Figure 4.1: A typical beam structure

Independent from the boundary conditions we will introduce later it will have to carry the load $q(x)$ by bending. That is, the bar displacement assumption is no more applicable and in this case we have to turn over to a displacement assumption which gives the structure some extended possibility to deform.

4.1 The Beam Displacement Assumption

Beam displacement assumption

Let us focus on a plane perpendicular to the x-axis and assume that this plane will remain flat. We now also give this plane the possibility to undergo both rotation and translation in the xy-plane as shown in figure 4.2. This is the so called *Beam displacement assumption*. This discussion will be limited to *plane bending*. That is the load $q(x)$ and displacement $v(x)$ will remain in the xy-plane. Note that this also requires a cross section which is symmetric around the z-axis. To include a general 3D deformation of a beam is straight forward as

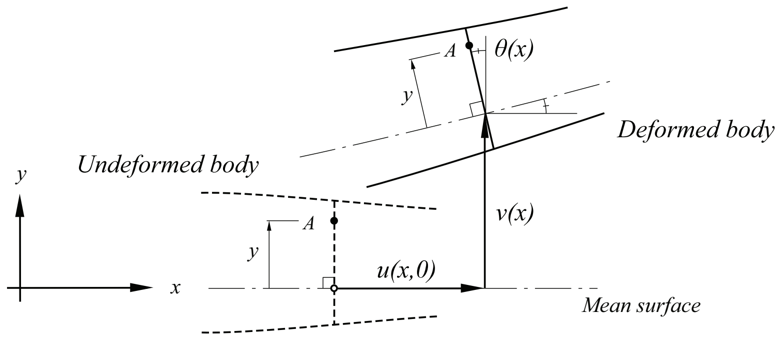


Figure 4.2: A typical beam deformation

Mean surface

long as the shear center and the center of gravity of the cross section coincide. The *Mean surface* is the plane which is exposed to neither tensile stresses nor to compressive stresses, if the beam carries a bending moment **only**. This also means that the fibers in this surface will keep its initial length and the strain is equal to zero. This mean surface plane always coincide with the xz-plane in the coordinate system.

4.2 The Local Equations

An expression for the horizontal displacement $u(x, y)$ in a general position A in a beam subjected to general loadings can be defined by the value of the y-axis and the slope of the deformed mean surface of the beam.

$$u(x, y) = u(x, 0) - y\frac{dv(x)}{dx} \tag{4.1}$$

This approximation is valid as long as θ is small (say less than 1 degree) and we have $\theta \simeq dv(x)/dx$. Calculation of the normal strain in the local x-direction ε_x from this displacement assumption is straight forward.

$$\varepsilon_x(x, y) = \frac{\partial u(x, y)}{\partial x} = \frac{du(x, 0)}{dx} - y\frac{d^2v(x)}{dx^2} \tag{4.2}$$

This is the compatibility relation in this *Euler-Bernoulli* beam theory displacement assumption. The physical interpretation of the first term is the strain generated by an axial force N carried by the beam.

As a constitutive relation we still use a linear elastic material and Hooke's 1D law applies. One can conclude from this that the stress σ_x and strain ε_x both varies as linear functions with respect to the y-direction.

$$\sigma_x(x,y) = E(x)\varepsilon_x(x,y) \tag{4.3}$$

For a balance law we once again use static equilibrium. First, the relationship between the bending moment $M(x)$ and the stress $\sigma_x(x,y)$, and the normal force $N(x)$ and the stress $\sigma_x(x,y)$ is established. From figure 4.3 we have for the

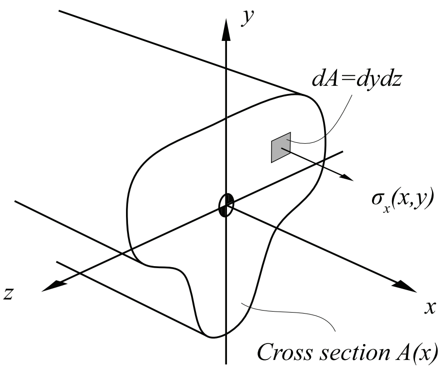


Figure 4.3: A single symmetric beam cross section

bending moment $M(x)$

$$M(x) = \int_{A(x)} y\sigma_x dA = \iint_{A(x)} y\sigma_x dydz \tag{4.4}$$

and for the axial normal force $N(x)$

$$N(x) = \int_{A(x)} \sigma_x dA = \iint_{A(x)} \sigma_x dydz. \tag{4.5}$$

Elimination of the stress and the strain by putting equations 4.3 and 4.2 into the equations 4.4 and 4.5 gives the following two equations

$$M(x) = E(x)\frac{du(x,0)}{dx} \underbrace{\iint_{A(x)} ydydz}_{=S(x)=0} - E(x)\frac{d^2v(x)}{dx^2} \underbrace{\iint_{A(x)} y^2dydz}_{=I(x)} \tag{4.6}$$

and

$$N(x) = \overbrace{E(x)\frac{du(x,0)}{dx}\underbrace{\iint_{A(x)} dydz}_{=A(x)}}^{=N(x)} - \frac{d^2v(x)}{dx^2}\underbrace{\iint_{A(x)} ydydz}_{=S(x)=0} \tag{4.7}$$

From this we can draw several important conclusions. First, we introduce the following closely related geometrical definitions for the cross section of the beam.

$$A(x) = \iint\limits_{A(x)} dydz; \qquad S(x) = \iint\limits_{A(x)} ydydz; \qquad I(x) = \iint\limits_{A(x)} y^2 dydz \tag{4.8}$$

These are the cross section $A(x)$, the first moment (statical moment) $S(x)$ and the area moment of inertia $I(x)$. Further on, we find that $S(x)$ must be equal to zero when this integral is taken over the entire cross section. The reason is that the mean surface of the beam always coincide with the center of gravity for the cross section. From equation 4.9 we then obtain

$$M(x) = -E(x)I(x)\frac{d^2v(x)}{dx^2} \tag{4.9}$$

which means that the bending moment is proportional to the curvature $\kappa \simeq d^2v(x)/dx^2$ of the beam for small displacements.

Concerning the bending stress $\sigma_x(x,y)$ in the beam an useful equation is received by putting equation 4.2 and 4.9 into equation 4.3

$$\sigma_x(x,y) = \frac{N(x)}{A(x)} + \frac{M(x)y}{I(x)} \tag{4.10}$$

Let us now study equilibrium for a short slice Δx of a generally loaded beam where vertical force and moment equilibrium equations give us

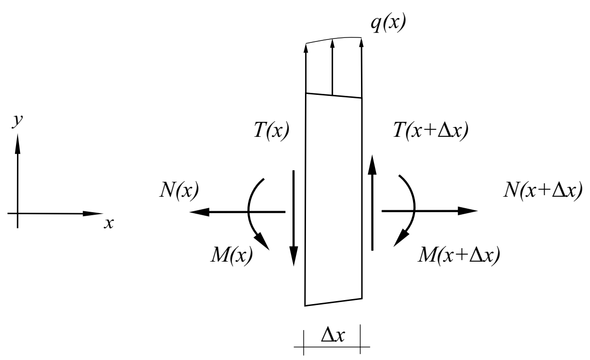


Figure 4.4: Forces and moments acting on a slice Δx of the beam

$$\uparrow \quad T(x+\Delta x) - T(x) + q(x)\Delta x = 0 \tag{4.11}$$

$$\curvearrowright \quad M(x+\Delta x) - M(x) + T\Delta x + q(x)\Delta x\frac{\Delta x}{2} = 0. \tag{4.12}$$

These equations can be simplified by use of Taylor's formula, thus

$$T(x+\Delta x) \approx T(x) + \frac{dT(x)}{dx}\Delta x \tag{4.13}$$

$$M(x+\Delta x) \approx M(x) + \frac{dM(x)}{dx}\Delta x. \tag{4.14}$$

Putting the equations 4.13 and 4.14 into equations 4.11 and 4.12 and letting Δx go to zero gives the following general equilibrium equations.

$$\frac{dT(x)}{dx} = -q(x) \tag{4.15}$$

$$\frac{dM(x)}{dx} = T(x) \tag{4.16}$$

By putting equations 4.9 and 4.15 into a derivative with respect to x of equation 4.16 we have eliminated both the bending moment $M(x)$ and the shear force $T(x)$ and we will end up with the well known

$$\frac{d^2}{dx^2}\left(E(x)I(x)\frac{d^2v(x)}{dx^2}\right) = q(x) \tag{4.17}$$

Elastic curve differential equation. This is a fourth-order differential equation in the transversal displacement $v(x)$ which is perpendicular to the mean surface of the beam. *Elastic curve*

In order to be able to solve this equation we need to apply boundary conditions. Four different boundary conditions, two at each end of the beam are required.

4.3 A Strong Formulation

After having a known set of boundary conditions it is possible to establish a well-posed strong formulation of this beam deflection problem. A large variety of the different combinations of boundary conditions are possible at least if we also include statically indeterminate beams into the discussion. When it comes to statically determinate systems there are mainly two different combinations. One can have a moment-free support at both ends or a rigid support at one end and the other end free.

Here we will use this very last case as an example for the discussion. That is, the boundary conditions are a known displacement $v(0) = g_1$ and a known rotation $dv(0)/dx = g_2$ at the left end and a given moment $M(L) = M_0 = h_1$

and a given vertical concentrated force $T(L) = T_0 = h_2$ acting in the right end. The left end of the interval will be given the notation S_g and the right end will be called S_h.

These four boundary conditions and the *Elastic curve* differential equation constitutes a well-posed fourth-order Boundary-Value problem $\mathbb{S}$.

Box: $\mathbb{S}$ ***'Strong form for an Euler-Bernoulli Beam'***

Given $q(x)$, h_1, h_2, g_1 and g_2. Find$v(x)$ such that

$$\frac{d^2}{dx^2}\left(E(x)I(x)\frac{d^2v(x)}{dx^2}\right) - q(x) = 0 \qquad \forall \quad x \in \Omega =]0, L[$$

$$v(0) = g_1 \quad \text{on } S_g \qquad -E(L)I(L)\frac{d^2v(L)}{dx^2} = h_1 \quad \text{on } S_h$$

$$\frac{dv(0)}{dx} = g_2 \quad \text{on } S_g \qquad -\frac{d}{dx}\left(E(L)I(L)\frac{d^2v(L)}{dx^2}\right) = h_2 \quad \text{on } S_h$$

Remarks:

- This formulation $\mathbb{S}$ is a Strong formulation of a linear static 1D Beam deflection problem and from a mathematical point of view this is a 1D fourth-order mixed *Boundary Value Problem*.

Boundary Value Problem

- $v(0) = g_1$ and $dv(0)/dx = g_2$ are *Essential* boundary conditions. If $g_1 \neq 0$ or $g_2 \neq 0$ the boundary condition is a non-homogeneous one.

Essential

- The total surface S consists in this 1D case only of the two end cross sections S_h and S_g.

- $h_1 = M_0$ and $h_2 = T_0$ are *Natural* boundary conditions.

Natural

- The boundary value problem is mixed because there are both essential and natural boundary conditions. Later on we will be aware of that some essential boundary conditions always have to exist to be able to guarantee the uniqueness of the solution of the matrix problem $\mathbb{M}$.

4.4 A Weak Formulation

Weight function

This Strong formulation can always be transferred into an equivalent Weak formulation by multiplication of an arbitrary *Weight function* $w(x)$ and integration over the domain Ω.

$$\int_0^L w(x)\left(\frac{d^2}{dx^2}\left(E(x)I(x)\frac{d^2v(x)}{dx^2}\right) - q(x)\right)dx = 0 \tag{4.18}$$

After partial integration of the first term we obtain

$$\left[w\frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\right]_0^L - \int_0^L \frac{dw}{dx}\frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)dx - \int_0^L wq dx = 0 \tag{4.19}$$

and after a second partial integration of this first term

$$\left[w\frac{d}{dx}\left(EI\frac{d^2v}{dx^2}\right)\right]_0^L - \left[\frac{dw}{dx}EI\frac{d^2v}{dx^2}\right]_0^L + \int_0^L \frac{d^2w}{dx^2}EI\frac{d^2v}{dx^2}dx - \int_0^L wq dx = 0. \tag{4.20}$$

Let us now introduction some specific requirements on the choice of weight functions $w(x)$. Make the choice from an infinite set $\mathcal{V}$ of functions where all members $w_i(x)$ must explore the following properties

$$\mathcal{V} = \{w_i(\cdot)|w_i(\cdot) = 0,\ w_i'(\cdot) = 0 \quad \text{on } S_g\} \tag{4.21}$$

where $w_i'(x) = dw(x)/dx$. By this restriction on the weight function two of six terms in equation 4.20 above will vanish and the following Weak formulation is summarized in box $\mathbb{W}$ below.

Box: $\mathbb{W}$ ***'Weak form for an Euler-Bernoulli Beam'***

Given $q(x),\ T_0,\ M_0,\ g_1$ and g_2. Find$v(x)$ such that

$$\int_0^L \frac{d^2w(x)}{dx^2}E(x)I(x)\frac{d^2v(x)}{dx^2}\,dx = \int_0^L w(x)q(x)\,dx + w(L)T_0 - \frac{dw(L)}{dx}M_0$$

$$v(0) = g_1 \qquad \frac{dv(0)}{dx} = g_2$$

Remarks:

- This weak formulation $\mathbb{W}$ will serve a base for applying a weighted residual method such as the Galerkin method.
- Two times partial integration is performed because that opens a possibility to later on end up in a symmetric system linear algebraic equations that is more efficiently solved in the computer compared to a non-symmetric one. It always reduces the requirements on regularity of the approximation of the unknown function $v(x)$.
- The natural boundary condition is implicitly contained in the integral equation.
- It can be shown that the Strong and the Weak formulations are equivalent.

4.5 A Galerkin Formulation

In all *Weighted Residual Methods* the approximation of the unknown function, in this case $v(x)$, is built up from a sum of *test* t_i functions and a known function t_0 taking care of non-homogeneous essential boundary conditions.

$$v(x) \approx v^h(x) = t_1(x)a_1 + \ldots + t_n(x)a_n + t_0(x) = \sum_{i=1}^{n} t_i a_i + t_0(x) \tag{4.22}$$

The *weight* function $w(x)$ can also be built up as sum of functions ϕ_i as follows

$$w(x) = \phi_1(x)c_1 + \phi_2(x)c_2 + \ldots + \phi_n(x)c_n = \sum_{i=1}^{n} \phi_i c_i \tag{4.23}$$

Shape functions

What is typical to a Galerkin formulation is, as we already have seen, that both the approximation and the weight function are composed from the same set of functions, so called *Shape functions* N_i where all N_i's belong to the set $\mathcal{V}$.

$$t_i(x) = \phi_i(x) = N_i(x) \tag{4.24}$$

where

$$\mathcal{V} = \{N_i(\cdot) | N_i(\cdot) = 0,\ N_i'(\cdot) = 0 \quad \text{on } S_g\} \tag{4.25}$$

A discrete Galerkin finite element formulation for beams can now be based on the following basic expressions

$$v(x) = \boldsymbol{N}(x)\boldsymbol{a} + \bar{N}_1(x)g_1 + \bar{N}_2(x)g_2 \tag{4.26}$$

$$w(x) = \boldsymbol{N}(x)\boldsymbol{c} \quad \text{or equivalently} \quad w(x) = \boldsymbol{c}^T\boldsymbol{N}^T(x) \tag{4.27}$$

where

$$\boldsymbol{N}(x) = \begin{bmatrix} N_1(x) & N_2(x) & \ldots & N_n(x) \end{bmatrix},\ \boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix},\ \boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}. \tag{4.28}$$

The functions $\bar{N}_1$ and $\bar{N}_2$ are **not** shape functions taking part in the overall approximation of the solution. These functions are only needed for taking care of non-homogeneous essential boundary conditions. That is, $g_1 \neq 0$ and $g_2 \neq 0$ and in cases where both these known values are equal to zero we can drop the two last terms. Later on we will se that the functions $\bar{N}_1$ and $\bar{N}_2$ will be closely related to the choice of shape functions N_i. But we require that these functions fulfill the following basic properties.

$$\bar{N}_1(0) = 1\,, \quad \frac{d\bar{N}_1(0)}{dx} = 0 \qquad \text{and} \qquad \bar{N}_2(0) = 0\,, \quad \frac{d\bar{N}_2(0)}{dx} = 1 \tag{4.29}$$

By putting this restrictions to these functions they are independent to each other and $\bar{N}_2$ doesn't influence the approximation and $\bar{N}_1$ doesn't influence the first derivative of the approximation $v^h(x)$.

Remarks:

- From the Weak formulation, after two times partial integration, we can conclude that both the test function and the weight function must fulfill the basic requirement that

$$\int_0^L \frac{d^2w(x)}{dx^2}\frac{d^2v(x)}{dx^2}\,dx < \infty \quad \Rightarrow \quad N_i \in \mathcal{C}^1 \tag{4.30}$$

  where the set $\mathcal{C}^1$ is all functions with at least a continuous **first-order** derivative.

- This means that a useful approximation of the beam deflection must have a continuous first-order derivative over the element borders. All *Hermitian* polynomial expressions have this property. *Hermitian*

Before we proceed by putting equations 4.26 and 4.27 into box $\mathbb{W}$ we pick up the opportunity to define the $\boldsymbol{B}$ matrix as follows

$$\frac{d^2v(x)}{dx^2} = \underbrace{\begin{bmatrix} \frac{d^2N_1(x)}{dx^2} & \dots & \frac{d^2N_n(x)}{dx^2} \end{bmatrix}}_{=\boldsymbol{B}} \boldsymbol{a} + \frac{d^2\bar{N}_1(x)}{dx^2}g_1 + \frac{d^2\bar{N}_2(x)}{dx^2}g_2 =$$

$$\boldsymbol{B}\boldsymbol{a} + \frac{d^2\bar{N}_1(x)}{dx^2}g_1 + \frac{d^2\bar{N}_2(x)}{dx^2}g_2 \tag{4.31}$$

$$\frac{d^2w(x)}{dx^2} = \boldsymbol{c}^T\boldsymbol{B}^T(x) \tag{4.32}$$

and we obtain the following discrete Galerkin formulation for beam problems.

Box: $\mathbb{G}$ ***'Galerkin form of an Euler-Bernoulli Beam'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{c}^T(\boldsymbol{K}\boldsymbol{a} - \boldsymbol{f}) = \boldsymbol{c}^T\boldsymbol{r} = 0$

for all choices of the vector $\boldsymbol{c}$ (the weight function)

where the global stiffness matrix $\boldsymbol{K}$ and the global load vector $\boldsymbol{f}$ are identified as follows

$$\boldsymbol{K} = \int_0^L \boldsymbol{B}^T(x)E(x)I(x)\boldsymbol{B}(x)\,dx \tag{4.33}$$

$$\boldsymbol{f} = \overbrace{\int_0^L \boldsymbol{N}^T(x)q(x)\,dx}^{=\boldsymbol{f}_d} + \overbrace{\boldsymbol{N}^T(L)T_0 - \frac{d\boldsymbol{N}^T(L)}{dx}M_0}^{=\boldsymbol{f}_h}$$
$$- \underbrace{\int_0^L \boldsymbol{B}^T(x)E(x)I(x)\left(\frac{d^2\bar{N}_1(x)}{dx^2}g_1 + \frac{d^2\bar{N}_2(x)}{dx^2}g_2\right)\,dx}_{=\boldsymbol{f}_g} = \boldsymbol{f}_d + \boldsymbol{f}_h - \boldsymbol{f}_g \tag{4.34}$$

4.6 A Matrix Formulation

Once again the conclusion from the Galerkin formulation is that for all choices of the vector $\boldsymbol{c}$ the scalar product with the unbalanced residual force vector $\boldsymbol{r}$ must be equal to zero.

$$\boldsymbol{c}^T\boldsymbol{r} = 0 \quad \Rightarrow \quad \boldsymbol{r} = \boldsymbol{K}\boldsymbol{a} - \boldsymbol{f} = \boldsymbol{0} \tag{4.35}$$

The only solution to this is that the force vector $\boldsymbol{r}$ is equal to a **zero vector** which means that equilibrium is achieved, measured at the nodal forces.

A matrix problem consisting of n linear algebraic equations can now be established.

Box: $\mathbb{M}$ ***'Matrix form of an Euler-Bernoulli Beam'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$

where $\boldsymbol{K}$ and $\boldsymbol{f}$ are known quantities

Remarks:

nodal rotations

- Concerning the global freedom vector $\boldsymbol{a}$ in this beam formulation it will, as we will understand from the next section, not only contain nodal displacements. It also includes the *nodal rotations*.
- After the number of elements and the nature of shape functions N_i has been decided it is straight forward to calculate both the matrix $\boldsymbol{K}$ and vector $\boldsymbol{f}$
- From a mathematical point of view, and exactly as in the bar formulation, this discussion can be summarized as

  $$\mathbb{L} \Rightarrow \mathbb{S} \Leftrightarrow \mathbb{W} \approx \mathbb{G} \Leftrightarrow \mathbb{M}$$

. Sources for errors in this mathematical model of reality are deviations from reality in the constitutive and the compatibility relations, deviations in the selected boundary conditions and numerical errors due to use of a limited number of Finite Elements with a specific behavior in each element.

- One can show that the solution to the matrix problem $\mathbb{M}$ always exists and has a unique solution if the global stiffness matrix $\boldsymbol{K}$ is non-singular. In this beam formulation we have to prevent rigid body motions such as translation in the y-direction and rotation around the y-axis. By doing so the global stiffness matrix $\boldsymbol{K}$ is positive definite and the following holds

$$\boldsymbol{a}^T \boldsymbol{K} \boldsymbol{a} > 0 \quad \forall \quad \boldsymbol{a} \neq \boldsymbol{0} \quad \Rightarrow \quad \det(\boldsymbol{K}) > 0$$

  Later on we will add axial stiffness from our 1D bar to this beam. Such an element will have 6 degrees of freedoms and this element is a complete 2D frame element. This means that we have to prevent rigid body motion in the global x-direction.

- In this 1D beam case the matrix $\boldsymbol{K}$ is symmetric with a close to diagonal population with a bandwidth of 4.

4.7 A 2D 2-node Beam Element

It is now time to try to find out what is the nature and the behavior of shape functions fulfilling what is required and postulated so far in this discussion. On the element level we have to be able to identify shape functions N_i^e and freedoms a_i^e associated to the nodes describing an approximation for the transversal displacements

$$v^e = N_1^e a_1^e + \ldots + N_m^e a_m^e = \begin{bmatrix} N_1^e & \ldots & N_m^e \end{bmatrix} \begin{Bmatrix} a_1^e \\ \vdots \\ a_m^e \end{Bmatrix} = \boldsymbol{N^e a^e} \tag{4.36}$$

where m is the total number of freedoms associated to **one** element. In the example to come we will very soon realize that $m = 4$. As we already have discussed, a proper shape function choice must belong to the set of functions $\mathcal{C}^1$, which consists of all functions with continuous first-order derivative.

From this the conclusion is that for each element we have to consider the translation in the y-direction and the slope at both ends of the element. In figure 4.5 an element with four freedoms is sketched. The element length L^e can be expressed in the element coordinates $L^e = x_{i+1} - x_i$. Let us now move focus over to a cubic polynomial expression such as

$$v(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 \tag{4.37}$$

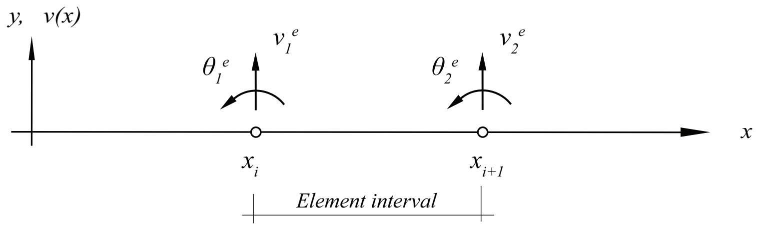


Figure 4.5: A 2-node beam element freedom set

which has four unknown constants α_1 to α_4. These constants can always be eliminated and expressed in the element freedom vector

$$\boldsymbol{a^e} = \left\{ \begin{array}{c} v_1^e \\ \theta_1^e \\ v_2^e \\ \theta_2^e \end{array} \right\}. \tag{4.38}$$

By performing the following steps it will be possible to establish all the element-local shape functions N_1^e to N_4^e by identifications. Let us start with just rewriting equation 4.37 in a matrix notation, as follows

$$v^e(x) = \left[\begin{array}{cccc} 1 & x & x^2 & x^3 \end{array} \right] \left\{ \begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \right\} = \boldsymbol{F}(x)\boldsymbol{\alpha} \tag{4.39}$$

and the slope $dv^e(x)/dx$ of the beam is

$$\theta(x) \simeq \frac{dv^e(x)}{dx} = \left[\begin{array}{cccc} 0 & 1 & 2x & 3x^2 \end{array} \right] \left\{ \begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \right\} \tag{4.40}$$

which also is the angle of rotation $\theta(x)$ for small angles. Let us now due to simplicity think of an element with its left end at $x_i = 0$ and the right end at $x_{i+1} = L^e$. By use of equations 4.39 and 4.40 for both ends the following four equations are obtained

$$\boldsymbol{a}^e = \left\{ \begin{array}{c} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{array} \right\} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 0 & 1 & 2L & 3L^2 \end{array} \right] \left\{ \begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{array} \right\} = \boldsymbol{A\alpha} \tag{4.41}$$

and we now have a relationship between the element freedom vector $\boldsymbol{a}^e$ and the unknown constants in the vector $\boldsymbol{\alpha}$. If the inverse $\boldsymbol{A}^{-1}$ exists it is straight forward to eliminate vector $\boldsymbol{\alpha}$ in-between equation 4.39 and equation 4.41 written as $\boldsymbol{\alpha} = \boldsymbol{A}^{-1}\boldsymbol{a}^e$ which gives

$$v^e(x) = \boldsymbol{F}(x)\boldsymbol{\alpha} = \boldsymbol{F}(x)\boldsymbol{A}^{-1}\boldsymbol{a}^e \tag{4.42}$$

and by comparing with equation 4.36 the element-local shape functions $\boldsymbol{N}^e$ can be identified as

$$\boldsymbol{N}^e(x) = \boldsymbol{F}(x)\boldsymbol{A}^{-1}. \tag{4.43}$$

After establishing the inverse $\boldsymbol{A}^{-1}$ the following

$$\boldsymbol{N}^e(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/L^{e^2} & -2/L^e & 3/L^{e^2} & -1/L^e \\ 2/L^{e^3} & 1/L^{e^2} & -2/L^{e^3} & 1/L^{e^2} \end{bmatrix} \tag{4.44}$$

multiplication will end up in the four element-local shape functions

$$\boldsymbol{N}^e(x) = \begin{bmatrix} N_1^e(x) & N_2^e(x) & N_3^e(x) & N_4^e(x) \end{bmatrix} \tag{4.45}$$

where

$$\begin{aligned} N_1^e(x) &= 1 - 3x^2/L^{e^2} + 2x^3/L^{e^3} \\ N_2^e(x) &= x - 2x^2/L^e + x^3/L^{e^2} \\ N_3^e(x) &= 3x^2/L^{e^2} - 2x^3/L^{e^3} \\ N_4^e(x) &= x^3/L^{e^2} - x^2/L^e \end{aligned} \tag{4.46}$$

which all are cubic expressions in x. In figure 4.6 these four shape functions are drawn. By putting equation 4.45 and 4.38 into 4.36 we obtain

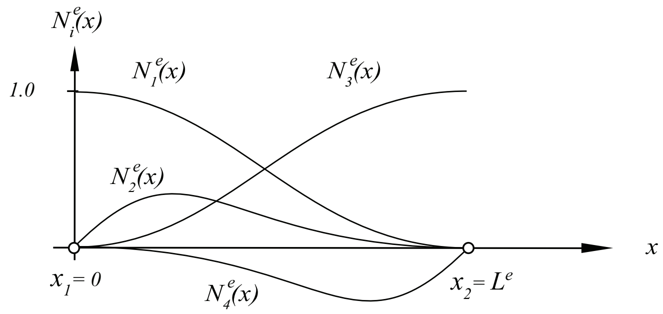


Figure 4.6: The element-local beam shape functions $N_1^e(x)$ to $N_4^e(x)$

$$v^e(x) = \boldsymbol{N}^e(x)\boldsymbol{a}^e = N_1^e(x)v_1^e + N_2^e(x)\theta_1^e + N_3^e(x)v_2^e + N_4^e(x)\theta_2^e \tag{4.47}$$

from which we can conclude that the displacement v^e at the nodes are unaffected by the value of the rotation. Concerning the slope or the rotation at the nodes we have in a similar fashion that the value of the displacement doesn't influence the rotation. This is exactly the property one should expect from a Hermitian polynomial approximation. This can be summarized as

$$\begin{aligned} N_1^e(x_1) = N_3^e(x_2) = \frac{dN_2^e(x_1)}{dx} = \frac{dN_4^e(x_2)}{dx} = 1 \\ \frac{dN_1^e(x_1)}{dx} = \frac{dN_3^e(x_2)}{dx} = N_2^e(x_1) = N_4^e(x_2) = 0 \end{aligned} \tag{4.48}$$

and the general conclusion is that the approximation of the displacement and the rotation **at the nodes** are uncoupled.

From equation 4.31 we can calculate the element-local $\boldsymbol{B}^e$ matrix

$$\boldsymbol{B}^e(x) = \begin{bmatrix} \frac{d^2N_1^e(x)}{dx^2} & \frac{d^2N_2^e(x)}{dx^2} & \frac{d^2N_3^e(x)}{dx^2} & \frac{d^2N_4^e(x)}{dx^2} \end{bmatrix} \tag{4.49}$$

where

$$\begin{aligned} \frac{d^2N_1^e(x)}{dx^2} &= B_1^e(x) = -6/L^{e^2} + 12x/L^{e^3} \\ \frac{d^2N_2^e(x)}{dx^2} &= B_2^e(x) = -4/L^e + 6x/L^{e^2} \\ \frac{d^2N_3^e(x)}{dx^2} &= B_3^e(x) = 6/L^{e^2} - 12x/L^{e^3} = -B_1^e \\ \frac{d^2N_4^e(x)}{dx^2} &= B_4^e(x) = 6x/L^{e^2} - 2/L^e \end{aligned} \tag{4.50}$$

which, as expected, all are linear functions in x. Now it is time to calculate the element stiffness matrix $\boldsymbol{K}^e$ and for a case with constant cross section properties we have

$$\boldsymbol{K}^e = \int_0^{L^e} \boldsymbol{B}^{e^T}(x) EI \boldsymbol{B}^e(x) dx = EI \int_0^{L^e} \begin{bmatrix} B_1^e \\ B_2^e \\ B_3^e \\ B_4^e \end{bmatrix} \begin{bmatrix} B_1^e & B_2^e & B_3^e & B_4^e \end{bmatrix} dx. \tag{4.51}$$

Expanding the integral into each position of the matrix product $\boldsymbol{B}^{e^T}\boldsymbol{B}^e$ only six different unique integrals are identified. This is because of the symmetry and use of the relation $B_1^e = -B_3^e$. Each of these integrals means integration of a second-order polynomial expression in x.

$$\boldsymbol{K}^e = EI \begin{bmatrix} \int_0^{L^e} B_1^{e^2} dx & \int_0^{L^e} B_1^e B_2^e dx & -\int_0^{L^e} B_1^{e^2} dx & \int_0^{L^e} B_1^e B_4^e dx \\ & \int_0^{L^e} B_2^{e^2} dx & -\int_0^{L^e} B_1^e B_2^e dx & \int_0^{L^e} B_2^e B_4^e dx \\ & & \int_0^{L^e} B_1^{e^2} dx & -\int_0^{L^e} B_1^e B_4^e dx \\ sym & & & \int_0^{L^e} B_4^{e^2} dx \end{bmatrix} \tag{4.52}$$

Let us look into details concerning only the first one which is

$$K_{11}^e = EI \int_0^{L^e} B_1^{e^2} dx = EI \int_0^{L^e} \left(-\frac{6}{L^{e^2}} + \frac{12x}{L^{e^3}} \right)^2 dx = \\ EI \left[\frac{48x^3}{L^{e^6}} + \frac{26x}{L^{e^4}} - \frac{72x^2}{L^{e^5}} \right]_0^{L^e} = 12 \frac{EI}{L^{e^3}} \tag{4.53}$$

and all the others are achieved similarly. The final result is the following 2D 2-node 4-freedom beam element. This element is useless as an element to be used in beam frame works because it do not resist any axial forces.

Box: *'2D beam element for transversal displacements only'*

$$\boldsymbol{K}^e = \frac{EI}{L^{e^3}} \begin{bmatrix} 12 & 6L^e & -12 & 6L^e \\ & 4L^{e^2} & -6L^e & 2L^{e^2} \\ & & 12 & -6L^e \\ sym & & & 4L^{e^2} \end{bmatrix}$$

This element is seldom implemented in commercial programs because it is only of academic interest to be used during hand calculations.

4.8 An Element Load Vector

The global load vector $\boldsymbol{f}$ has been split into 3 different parts

$$\boldsymbol{f} = \boldsymbol{f}_d + \boldsymbol{f}_h - \boldsymbol{f}_g \tag{4.54}$$

where the part $\boldsymbol{f}_d$ consists of contributions from the distributed load $q(x)$ which has to be evaluated at the element level and is given as follows

$$\boldsymbol{f}_d = \sum_{i=1}^{n} \boldsymbol{C}_i^{e^T} \underbrace{\int_{x_i}^{x_{i+1}} \boldsymbol{N}^{e^T}(x) q(x)\, dx}_{=\boldsymbol{f}_i^e} . \tag{4.55}$$

where $\boldsymbol{C}_i^{e^T}$ is a Boolean matrix. This matrix defines to which global freedom each of the element-local freedoms belongs. (For further details see the Bar chapter)

The element load vector $\boldsymbol{f}_i^e$ can be analyzed further if we assume that the given load $q(x)$ at most varies as a linear function over each element interval. That is the following linear expression

$$q(x) = (1 - \frac{x}{L^e})q_1^e + \frac{x}{L^e}q_2^e \tag{4.56}$$

is useful where q_1^e and q_2^e are the intensity of load $q(x)$ at the nodes. This idea means actually an "approximation" of given data of the problem which at the first glance seem to be a bit stupid. What we gain from this idea is that we can proceed one further step analytically and the input to the numerical procedure will be the load intensity at the nodes only. From an engineering point of view one can also argue that if the given load $q(x)$ varies rapidly with respect to x we need to use shorter elements anyway, if we would like to try to catch details caused by the rapid load change.

The element load vector $\boldsymbol{f}_i^e$ can be rewritten as

$$\boldsymbol{f}_i^e = \int_0^{L^e} \left\{ \begin{array}{c} 1 - 3x^2/L^{e^2} + 2x^3/L^{e^3} \\ x - 2x^2/L^e + x^3/L^{e^2} \\ 3x^2/L^{e^2} - 2x^3/L^{e^3} \\ x^3/L^{e^2} - x^2/L^e \end{array} \right\} \Big((1 - \frac{x}{L^e})q_1^e + \frac{x}{L^e}q_2^e\Big)\, dx \tag{4.57}$$

where we have four different integrals to solve. The polynomial expressions are all at most of a fourth order degree. The result from these four integrations gives

$$\boldsymbol{f}_i^e = \left\{ \begin{array}{c} L^e(7q_1^e + 3q_2^e)/20 \\ L^{e^2}(q_1^e/20 + q_2^e/30) \\ L^e(3q_1^e + 7q_2^e)/20 \\ -L^{e^2}(q_1^e/30 + q_2^e/20) \end{array} \right\} \quad \boldsymbol{f}_i^e = \left\{ \begin{array}{c} q_0 L^e/2 \\ q_0 L^{e^2}/12 \\ q_0 L^e/2 \\ -q_0 L^{e^2}/12) \end{array} \right\} \tag{4.58}$$

where the second expression is the special case when the load $q(x)$ is a constant, that is $q_1^e = q_2^e = q_0$. From this we can conclude that a line load $q(x)$ also generate moments as well as nodal loads.

In this context one should also mention that some people argue for an ad hoc reduced element load vector without nodal moments which sometimes also is implemented in commercial programs. The treatment shown here is called a *Consistent element load vector* and this is what comes out from this mathematical formulation.

Consistent element load vector

Concerning the boundary condition terms $\boldsymbol{f}_h$ and $\boldsymbol{f}_g$ we have

$$\boldsymbol{f}_h = \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ T_0 \\ -M_0 \end{Bmatrix} \qquad \boldsymbol{f}_g = \begin{Bmatrix} 12EIg_1/L^{e^3} + 6EIg_2/L^{e^2} \\ 6EIg_1/L^{e^2} + 4EIg_2/L^e \\ -12EIg_1/L^{e^3} - 6EIg_2/L^{e^2} \\ 6EIg_1/L^{e^2} + 4EIg_2/L^e \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{4.59}$$

where these are two global load vector contributions with the same number of rows as in the global unknown freedom vector $\boldsymbol{a}$. Please observe that the dimension of the terms alters between force and moment and in this discussion the essential boundary conditions occur at the left end of the beam and the natural bc. at the right end. Several other combinations are also of course possible.

4.9 A 2D Beam Element with Axial Stiffness

In most real space frame structures it is also necessary to take axial deformations into account. This is, as already mentioned, not included in the previous element. Fortunately it is easy to cure this deficiency of the element. In cases with small deformations it is a good approximation saying that deformations in the axial and the transversal directions evolves **independent** of each other.

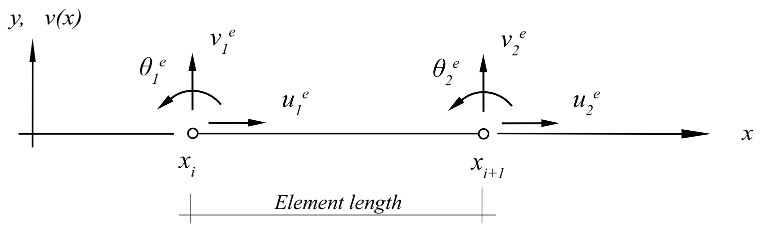


Figure 4.7: A 2D 2-node frame element freedom set

That is the bar problem and the beam problem can be solved independent from each other using the same finite element mesh. Such a beam element resisting axial stiffness also is often called a *Frame element* *Frame element*

Box: *'2D 2-node frame element'*

$$
\boldsymbol{K}^e = \frac{1}{L^e}\begin{bmatrix} EA & 0 & 0 & -EA & 0 & 0 \\ & 12EI/L^{e^2} & 6EI/L^e & 0 & -12EI/L^{e^2} & 6EI/L^e \\ & & 4EI & 0 & -6EI/L^e & 2EI \\ & & & EA & 0 & 0 \\ & & & & 12EI/L^{e^2} & -6EI/L^e \\ sym & & & & & 4EI \end{bmatrix}
$$

Please observe that the sequence between the six freedoms in the element is not critical. This selection is only one of several others which will work equally well. Most often textbooks starts, as has been done here, with the axial freedom u_1^e and continue with the two other freedoms at the first node v_1^e and θ_1^e and end up with the same freedoms at the second node of the element.

Still this element lacks one important feature to be a candidate for implementation in a general purpose finite element program. This 2D element stiffness matrix is established along a local direction and in a real space frame application one most likely will find members running in arbitrary directions in 2D or 3D space. This defect will be cured in the next section where we will discuss a 3D beam element in an arbitrary direction in 3D space.

4.10 A 3D Space Frame Element

To establish a general 3D beam element for any space frame analysis is rather straightforward. To add bending out of the plane it is just a matter of superimposing of a second load system bending the beam in the xy-plane. For an applied

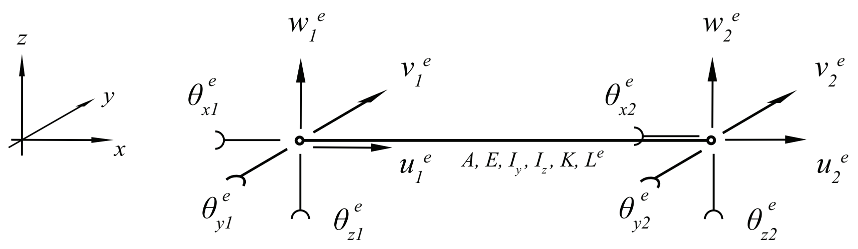


Figure 4.8: A 3D 12-freedom beam element defined in a local system

torque trying to twist the beam the obvious choice is to use St. Venant torsional theory. An element stiffness matrix $\boldsymbol{K}^e$ can now easily be establish from what we already know, as follows

$$
\boldsymbol{K}^e = \begin{bmatrix}
a_1 & 0 & 0 & 0 & 0 & 0 & -a_1 & 0 & 0 & 0 & 0 & 0 \\
 & b_1 & 0 & 0 & 0 & b_2 & 0 & -b_1 & 0 & 0 & 0 & b_2 \\
 & & c_1 & 0 & -c_2 & 0 & 0 & 0 & -c_1 & 0 & -c_2 & 0 \\
 & & & d_1 & 0 & 0 & 0 & 0 & 0 & -d_1 & 0 & 0 \\
 & & & & c_3 & 0 & 0 & 0 & c_2 & 0 & c_4 & 0 \\
 & & & & & b_3 & 0 & -b_2 & 0 & 0 & 0 & b_4 \\
 & & & & & & a_1 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & b_1 & 0 & 0 & 0 & -b_2 \\
 & & & & & & & & c_1 & 0 & c_2 & 0 \\
 & & & & & & & & & d_1 & 0 & 0 \\
 & s & y & m. & & & & & & & c_3 & 0 \\
 & & & & & & & & & & & b_3
\end{bmatrix} \tag{4.60}
$$

where

$$
\begin{array}{llll}
a_1 = EA/L^e & d_1 = GK/L^e & & \\
b_1 = 12EI_z/L^{e^3} & b_2 = 6EI_z/L^{e^2} & b_3 = 4EI_z/L^e & b_4 = 2EI_z/L^e \\
c_1 = 12EI_y/L^{e^3} & c_2 = 6EI_y/L^{e^2} & c_3 = 4EI_y/L^e & c_4 = 2EI_y/L^e
\end{array}
$$

and G is the torsional modulus of the material and K is the torsional proportional constant for the cross section.

Remarks:

- This element works superior with respect to classical beam theory as long as the cross section has two axes of symmetry.
- In cases with open thin-walled cross sections warping of the cross section can be significant and one should be careful. This typically happens in cross sections where the centroid and the shear center of the cross section do not coincide.
- The selected sequence between the element freedoms $\boldsymbol{a}^e$ is in this case

$$
\boldsymbol{a}^{e^T} = \{ \; u_1^e \quad v_1^e \quad w_1^e \quad \theta_{x1}^e \quad \theta_{y1}^e \quad \theta_{z1}^e \quad u_2^e \quad v_2^e \quad w_2^e \quad \theta_{x2}^e \quad \theta_{y2}^e \quad \theta_{z2}^e \; \}
$$

- One useful technique for overcoming this problem is to use shell element for modeling of such cross section.

We now need an element which can be used in an arbitrary direction in 3D space. This can be taken care of by the rules for transformation of a vector in 3D space. Please note that the previous coordinate system in figure 4.8 now has become a prime coordinate system or local coordinate system and the element freedoms present in figure 4.9 are not the same as the ones in figure 4.8.

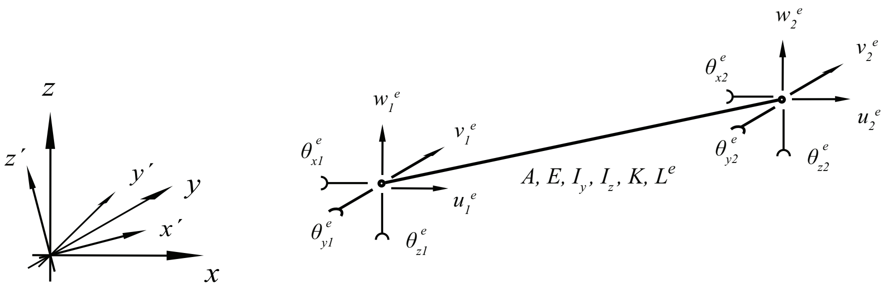


Figure 4.9: A 3D **global** 12-freedom beam element

Let us focus on a vector $\boldsymbol{u}$ defined by components in the global coordinate system and the same vector now called $\boldsymbol{u'}$ defined by components in the local coordinate system. The relation between the components of this vectors is

$$\boldsymbol{u'} = \boldsymbol{\Lambda u} \tag{4.61}$$

where the matrix $\boldsymbol{\Lambda}$ is 3x3 matrix containing direction cosines between the axes of the two systems. The transformation of the element freedom vector $\boldsymbol{a'}^e$ in the local system to element freedoms $\boldsymbol{a}^e$ defined in the global system is given by

$$\boldsymbol{a'}^e = \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda} \end{bmatrix} = \boldsymbol{T}\boldsymbol{a}^e \tag{4.62}$$

and this can can be used for transformation of the element-local equilibrium equation

$$\boldsymbol{K'}^e \boldsymbol{a'}^e = \boldsymbol{f'}^e. \tag{4.63}$$

where the matrix $\boldsymbol{K'}^e$ is equivalent to what is called $\boldsymbol{K}^e$ in equation 4.60. By putting equation 4.62 into equation 4.63 and multiplying with $\boldsymbol{T}^T$ from the left we can identify a global 3D element stiffness matrix $\boldsymbol{K}^e$ and a global element load vector $\boldsymbol{f}^e$ for a beam or frame element resisting axial forces and twisting moment and therefore useful in general 3D space frame analysis.

$$\underbrace{\boldsymbol{T}^T \boldsymbol{K'}^e \boldsymbol{T}}_{=\boldsymbol{K}^e} \boldsymbol{a}^e = \underbrace{\boldsymbol{T}^T \boldsymbol{f'}^e}_{=\boldsymbol{f}^e} \tag{4.64}$$

4.11 Stress and Strain Calculations

As for bar elements and all other elements based on basically the same finite element formulation, the strain and the stress is calculated element by element at the end of the numerical analysis. We have by generalizing equation 4.2 and use of equation 4.3

$$\sigma_x(x,y,z) = E\varepsilon_x(x,y,z) = E\left(\frac{du(x,0)}{dx} - y\frac{d^2v(x)}{dx^2} - z\frac{d^2w(x)}{dx^2}\right) \tag{4.65}$$

where the displacement component in the local y-direction is $w(x)$ and $v(x)$ is the displacement in the y-direction. The first term is the axial stress due to a force acting along the local direction of the beam. The axial and the bending stresses together can be expressed in the element freedom vector as follows

$$\begin{aligned}\sigma_x(x,y,z) = E\Bigg(&\frac{1}{L^e}\left[\begin{array}{cc} -1 & 1 \end{array}\right]\left\{\begin{array}{c} u_1^e \\ u_2^e \end{array}\right\} \\ &- y\left[\begin{array}{cccc} B_1^e & B_2^e & B_3^e & B_4^e \end{array}\right]\left\{\begin{array}{c} v_1^e \\ \theta_{z1}^e \\ v_2^e \\ \theta_{z2}^e \end{array}\right\} \\ &- z\left[\begin{array}{cccc} B_1^e & B_2^e & B_3^e & B_4^e \end{array}\right]\left\{\begin{array}{c} w_1^e \\ \theta_{y1}^e \\ w_2^e \\ \theta_{y2}^e \end{array}\right\}\Bigg)\end{aligned} \tag{4.66}$$

This expression is used in the **TRINITAS** program for plotting of the stress level on the surface of every beam element.

4.12 Numerical Examples

A 2D console beam

Let us consider a simple standard case which also has an analytical solution. The left end of the beam has a completely rigid support and the right end is free. A constant load $q(x) = q_0$ per unit length acts along the entire length of the beam. At the right end acts a concentrated force P. That is, from a comparison with box $\mathbb{S}$ we have the following values of the boundary conditions $g_1 = g_2 = h_1 = 0$ and $h_2 = P$.

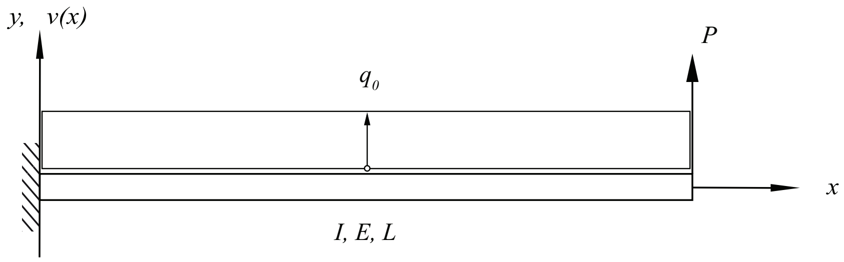


Figure 4.10: A 2D console beam example

The exact solution to this problem is received from the *Elastic curve* equation 4.17 which in this case can be simplified as follows

$$\frac{d^4v(x)}{dx^4} = q_0/EI. \tag{4.67}$$

because both the cross section geometry and the material do not change along the beam. After integrating four times without boundaries we have

$$\frac{d^3v(x)}{dx^3} = q_0x/EI + C_1$$

$$\frac{d^2v(x)}{dx^2} = q_0x^2/2EI + C_1x + C_2$$

$$\frac{dv(x)}{dx} = q_0x^3/6EI + C_1x^2/2 + C_2x + C_3$$

$$\underline{v(x) = q_0x^4/24EI + C_1x^3/6 + C_2x^2/2 + C_3x + C_4}.$$

Four unknown constants C_1 to C_4 have to be determined from the given boundary conditions. The left end conditions $g_1 = g_2 = 0$ imply immediately that $C_3 =$

$C_4 = 0$. The conditions at the right end gives

$$h_1 = 0 \quad \Rightarrow \quad \frac{d^2v(L)}{dx^2} = q_0L^2/2EI + C_1L + C_2 = 0$$

$$h_2 = P \quad \Rightarrow \quad \frac{d^3v(L)}{dx^3} = q_0L/EI + C_1 = -P/EI$$

C_1 and C_2 equal to

$$C_1 = -P/EI - q_0L/EI \qquad C_2 = PL/EI + q_0L/2^2EI$$

and all constants are known and the solution $v(x)$ is

$$\underline{v(x) = \frac{q_0L^4}{24EI}\left(6(\frac{x}{L})^2 - 4(\frac{x}{L})^3 + (\frac{x}{L})^4\right) + \frac{PL^3}{6EI}\left(3(\frac{x}{L})^2 - (\frac{x}{L})^3\right).}$$

Let us now solve this problem numerically by Finite Elements and hand calculations. Use only one element with just a vertical displacement v and a rotation θ in the right end. This one element problem with its left end fixed will end up in the following two global stiffness equations describing the behavior of the console beam.

$$\frac{EI}{L^3}\begin{bmatrix} 12 & -6L \\ -6L & 4L^2 \end{bmatrix}\begin{Bmatrix} v \\ \theta \end{Bmatrix} = \begin{Bmatrix} q_0L/2 + P \\ -q_0L^2/12 \end{Bmatrix}$$

The two global freedoms v and θ can be calculated as follows

$$\begin{Bmatrix} v \\ \theta \end{Bmatrix} = \frac{1}{12L^2}\begin{bmatrix} 4L^2 & 6L \\ 6L & 12 \end{bmatrix}\frac{L^3}{EI}\begin{Bmatrix} q_0L/2 + P \\ -q_0L^2/12 \end{Bmatrix} \qquad \Rightarrow$$

$$\begin{Bmatrix} v \\ \theta \end{Bmatrix} = \begin{Bmatrix} q_0L^4/8EI + PL^3/3EI \\ q_0L^3/6EI + PL^2/2EI \end{Bmatrix}.$$

From this result we can conclude that both the displacement $v(x = L)$ and the rotation $\theta = dv(x = L)/dx$ at the right end are exactly the same as from the analytical solution above. Let us also compare the displacements along the beam at an arbitrary x-value. By use of the values of the freedoms v and θ above putted into the equations 4.36 and 4.46 we obtain

$$v^e(x) = N_3^e(x)v + N_3^e(x)\theta = \left(3(\frac{x}{L})^2 - 2(\frac{x}{L})^3\right)\left(\frac{q_0L^4}{8EI} + \frac{PL^3}{3EI}\right) + \left((\frac{x}{L})^3 - (\frac{x}{L})^2\right)L\left(\frac{q_0L^3}{6EI} + \frac{PL^2}{2EI}\right)$$

which after simplification gives

$$v(x) = \frac{q_0L^4}{24EI}\left(5(\frac{x}{L})^2 - 2(\frac{x}{L})^3\right) + \frac{PL^3}{6EI}\left(3(\frac{x}{L})^2 - (\frac{x}{L})^3\right).$$

A second important conclusion which can be drawn from this results is that the finite element approximation in the interior of the element is exact for the part of the displacement generated by the concentrated load P but not for the part emanating from the distributed load q_0. This is obvious and this conclusion could be drawn as a cubic polynomial was used for the beam shape function which is **not** sufficient for catching the fourth order term originating from the particular solution to the *elastic curve* equation for this example.

That is, in linear static elastic beam models it is enough using just one element for intervals **without** any distributed load.

If we would like to do a numerical comparison with a finite element program we need numerical values. Use $L = 1.0\ m$, $P = 1000.\ N$, $q_0 = 1000.\ N/m$, $E = 2.0 \cdot 10^{11}\ Pa$. Assume that the cross section has a height h and a width b where $b = 4h = 0.01\ m$ which gives the area moment of inertia $I = 5.333 \cdot 10^{-8}$ m^4. At the tip of the beam we have the best agreement between this analytical solution and a finite element solution and from the two load systems we receive

$$v(x = L) = 0.01172 + 0.03125 \approx 0.043m$$

from both the analytical and a finite element solution with only one element.

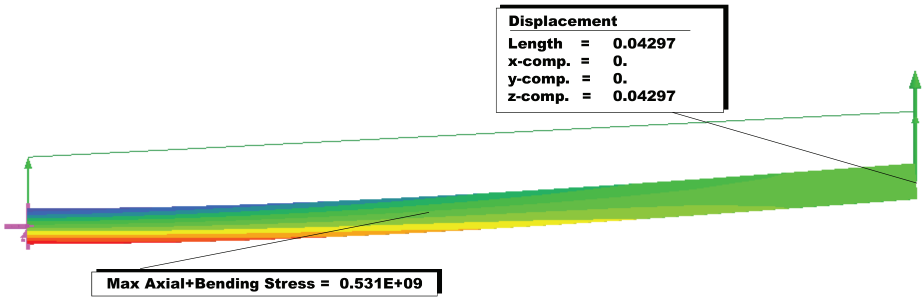


Figure 4.11: A one element **TRINITAS** analysis

4.13 Summary

After assuming that every cut perpendicular to the mean surface of the beam will remain plane the following local equations can be stated.

Box: $\mathbb{L}$ ***'Local Euler-Bernoulli Beam Equations'***

$$\frac{d^2M(x)}{dx^2} + q(x) = 0, \quad M(x) = \iint\limits_{A(x)} y\sigma_x(x,y)dydz$$

$$\sigma_x(x,y) = E(x)\varepsilon_x(x,y)$$

$$\varepsilon_x(x,y) = \frac{du(x,0)}{dx} - y\frac{d^2v(x)}{dx^2}$$

By elimination of the moment $M(x)$, the strain $\varepsilon_x(x,z)$ and the stress $\sigma_x(x,z)$ the following 1D fourth-order Boundary-Value problem is received.

Box: $\mathbb{S}$ ***'Strong form for an Euler-Bernoulli Beam'***

Given $q(x),\ h_1,\ h_2,\ g_1$ and g_2. Find$v(x)$ such that

$$\frac{d^2}{dx^2}\left(E(x)I(x)\frac{d^2v(x)}{dx^2}\right) - q(x) = 0 \qquad \forall \quad x \in \Omega =]0, L[$$

$$v(0) = g_1 \quad \text{on } S_g \qquad -E(L)I(L)\frac{d^2v(L)}{dx^2} = h_1 \quad \text{on } S_h$$

$$\frac{dv(0)}{dx} = g_2 \quad \text{on } S_g \qquad -\frac{d}{dx}\left(E(L)I(L)\frac{d^2v(L)}{dx^2}\right) = h_2 \quad \text{on } S_h$$

The following equivalent weak formulation is established after partial integrations and restricting the weight function to be equal to zero where essential boundary conditions are present.

Box: $\mathbb{W}$ ***'Weak form for an Euler-Bernoulli Beam'***

Given $q(x),\ T_0,\ M_0,\ g_1$ and g_2. Find$v(x)$ such that

$$\int_0^L \frac{d^2w(x)}{dx^2}E(x)I(x)\frac{d^2v(x)}{dx^2}\,dx = \int_0^L w(x)q(x)\,dx + w(L)T_0 - \frac{dw(L)}{dx}M_0$$

$$v(0) = g_1 \qquad \frac{dv(0)}{dx} = g_2$$

By requiring that both the approximation and the weight function are built up from the same set of functions a discrete Galerkin formulation is achieved.

Box: $\mathbb{G}$ ***'Galerkin form of an Euler-Bernoulli Beam'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{c}^T(\boldsymbol{K}\boldsymbol{a} - \boldsymbol{f}) = \boldsymbol{c}^T\boldsymbol{r} = 0$

for every choice of the vector $\boldsymbol{c}$ (the weight function)

The single equation in the Galerkin formulation above, which shall be interpreted as a scalar product between a row vector $\boldsymbol{c}^T$ and a column vector $\boldsymbol{r}$, will now turn over into a system of linear algebraic equations ($\mathbb{M}$) because the residual vector $\boldsymbol{r}$ must be equal to a zero vector.

Box: $\mathbb{M}$ ***'Matrix form of an Euler-Bernoulli Beam'***

Find $\boldsymbol{a}$ such that

$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$

where $\boldsymbol{K}$ and $\boldsymbol{f}$ are known quantities

In this beam case discussion the solution vector $\boldsymbol{a}$ to this matrix problem will contain both displacements and rotations at the nodes.

From here the numerical procedure starts and the following work flow can be identified.

Numerical Work Flow:

- Split the entire domain into a number of finite elements
- Define domain properties such as the Young's modulus E, the cross section A and the area moment of inertia I in the elements
- Define essential boundary conditions such as fixed or prescribed node displacements and/or rotations
- Define natural boundary condition as given concentrated or distributed forces or moment loads
- Make a global numbering sequence of all involved unknown freedoms (Normally done automatically by the program)

- Calculate all element stiffness matrices $\boldsymbol{K}_i^e$ and expand and add these stiffness coefficients into the appropriate positions in the global stiffness matrix $\boldsymbol{K}$
- Calculate all element load vectors $\boldsymbol{f}_i^e$ and expand and add these load contributions into the appropriate positions in the global load vector $\boldsymbol{f}$
- Solve the system of linear algebraic equation $\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f}$ by some Gauss' elimination or LR-factorization look-a-like procedure. (Further discussions concerning how to calculate the unknown vector $\boldsymbol{a}$ will be given later on in the chapters to come)
- Pick up the element freedom vector $\boldsymbol{a}_i^e$ from the global one $\boldsymbol{a}$ for one element at the time and calculate the strains and the stresses
- Investigate the results, hopefully in terms of a nice color picture showing the deformed structure with the stress levels in color, and try to examine the relevance of the achieved approximation.

In most finite element analyses, after the results have been accepted from an overall engineering point of view, one also have to accept the analysis from a numerical point of view which normally means a refinement of the mesh in some critical part of the domain trying to find out if the accuracy of the numerical results is sufficient.

Finally, one should not forget that the reason for doing the finite element analysis was some overall engineering question concerning how to design a certain piece of equipment and the finite element analysis only gives some hints concerning the size of displacements, strains and stresses in a **model of reality**.