

From Weighted Residual Methods to Finite Element Methods

Lars-Erik Lindgren

2014

TABLE OF CONTENT

<u>1 INTRODUCTION</u>	3
<u>2 DEFINITIONS</u>	4
<u>3 SHORT FINITE ELEMENT THEORY</u>	5
3.1 GALERKIN FORMULATION OF FINITE ELEMENT METHODS	5
3.2 RITZ FORMULATION OF FINITE ELEMENT METHODS	6
3.3 DISCRETISATION IN SPACE	6
3.4 TONTI DIAGRAMS AS A MEAN TO ILLUSTRATE RELATIONS	7
<u>4 WEIGHTED RESIDUAL METHODS</u>	11
4.1 SUBDOMAIN METHOD	12
4.2 COLLOCATION METHOD	12
4.3 LEAST-SQUARES METHOD	12
4.4 METHOD OF MOMENTS	13
4.5 GALERKIN METHOD	13
4.6 GALERKIN METHOD INCLUDING NATURAL BOUNDARY CONDITION IN RESIDUAL	13
4.7 PETROV-GALERKIN METHOD	13
4.8 COMPARISON OF WRM METHODS	14
4.8.1 PROBLEM DEFINITION AND EXACT SOLUTION	14
4.8.2 TRIAL FUNCTION AND RESIDUALS	14
4.8.3 SUBDOMAIN WEIGHT FUNCTIONS	15
4.8.4 COLLOCATION WEIGHT FUNCTIONS	18
4.8.5 WEIGHT FUNCTIONS IN THE LEAST SQUARES METHOD	19
4.8.6 METHOD OF MOMENTS WEIGHT FUNCTIONS	20
4.8.7 GALERKIN WEIGHT FUNCTIONS	21
4.8.8 GALERKIN METHOD INCLUDING NATURAL BOUNDARY CONDITION IN RESIDUAL	22
4.8.9 SUMMARY OF COMPARISONS	23
4.9 EXERCISES	23
<u>5 CLASSICAL AND COMPUTATIONAL GALERKIN METHODS</u>	25
5.1 EXERCISE	25
<u>6 FINITE ELEMENT METHODS</u>	26
6.1 GLOBAL WEIGHT AND TRIAL FUNCTIONS	28
6.2 NODAL BASED TRIAL AND WEIGHT FUNCTIONS	29
6.3 ASSEMBLY PROCEDURE	34
6.4 ELEMENT BASED TRIAL AND WEIGHT FUNCTIONS	35
6.5 PROCESSING OF FINITE ELEMENT RESULTS	42
6.6 EXERCISES	44
<u>7 NUMERICAL INTEGRATION</u>	49
<u>8 AXISYMMETRIC PROBLEMS</u>	51
<u>9 HIGHER ORDER DIFFERENTIAL EQUATIONS</u>	55
9.1 BERNOULLI BEAM	55
9.2 TIMOSHENKO BEAM	58
9.3 CANTILEVER BEAM PROBLEM	62

<u>10</u>	<u>CONVERGENCE OF FINITE ELEMENT FORMULATIONS</u>	<u>67</u>
<u>11</u>	<u>ISOPARAMETRIC MAPPING IN TWO DIMENSIONS</u>	<u>68</u>
<u>12</u>	<u>NAVIER'S EQUATION IN TWO DIMENSIONS</u>	<u>72</u>
12.1	DISPLACEMENTS AND GEOMETRY OF FOUR-NODE PLANE ELEMENT	73
12.2	STRAINS OF FOUR-NODE PLANE ELEMENT	75
12.3	CONSTITUTIVE MODEL FOR PLANE STRESS	76
12.4	FLOW CHART FOR CALCULATION OF FOUR NODE, PLANE STRESS ELEMENT	76
<u>13</u>	<u>CONVECTIVE HEAT TRANSFER – NON-SYMMETRIC PHYSICS</u>	<u>78</u>
<u>14</u>	<u>REFERENCES</u>	<u>83</u>
<u>15</u>	<u>APPENDIX.</u>	<u>1</u>
15.1	TENSOR AND VECTOR NOTATIONS IN MECHANICAL PROBLEMS	1
15.2	GALERKIN AND RITZ METHODS	2
15.3	CONDENSATION OF A SYSTEM OF EQUATIONS.	3
15.4	BERNOULLI BEAM	3
15.5	TIMOSHENKO BEAM	7
15.6	ONE-DIMENSIONAL ELEMENT FOR HEAT CONDUCTION	10

1 Introduction

The Finite Element Method (FEM) is a general method for solving partial differential equations of different types. The author has mostly used it to solve thermo-mechanical problems for manufacturing processes. The method has become a standard method in industry and has to a large extent replaced experiments and testing for quick evaluation of different design options. However, the use of computing power cannot replace the need for brain power. The citation below from [1] makes this clear;

“The (FEM) tools have become so easy to use, little thought is required to develop stress contours on parts with complex geometry. With little thought comes little chance of accuracy.” p. 7 in [1].

The purpose of the current text is to elucidate how finite element formulations for a given mathematical problem can be done. Most of the fundamentals of finite element formulations are easier to follow when applied to one-dimensional problems. Therefore this is the focus in the book. There are some extra complications when implementing the method for multidimensional problems and therefore this is included although this is also available in standard textbooks about FEM and therefore is only introduced in the current text.

A practical approach, rather than a mathematical, is taken in the book and therefore derivations but no complete proofs are given. There is a very large number of textbooks covering the finite element theory on basic as well as advanced levels. Its will be too an extensive list to include here. The reader is referred to the web – just google!¹

The early experience of the author from teaching FEM from courses with about 60 undergraduate students in mechanical engineering using standard textbooks covering FEM and its theory for static, linear elastic problems can be summarised as

- ten asked ‘Why do we need all this theory?’,
- then there was the quiet majority and
- ten percent asked ‘Can we program an element?’

This compendium is written for the last group with the hope that they enjoy the idea of understanding how FEM works.

The author believes that also anyone that wants to master the use of finite element modelling for nonlinear problems should have a thorough understanding of FEM. The method is best absorbed via learning by doing. Thus the students are given assignments to derive finite element formulations as well as implement them in MatlabTM.

¹ http://hogwarts.ucsd.edu/~pkrysl/faesor/faesor_publish.html - a MatlabTM - object oriented FE-code.

http://m2matlabdb.ma.tum.de/files.jsp?MC_ID=1&SC_ID=1 - Link to MatlabTM FEA-codes.

<http://www.ilsb.tuwien.ac.at/links/felinks.html> , <http://thcentral.com/finiteelement.htm> - web sites for further surfing.

2 Definitions

The most important definition is *model* – a symbolic device built to simulate and predict aspects of behaviour of a system. The word ‘aspects’ indicates that there is a limited, specific purpose for which the model is created. It is the *scope* of the model. Determining the scope is the most important step in the modelling process. What information is wanted? Why should the analysis be done? The scope determines what tool and model can be used. The scope determines together with ‘when’ the analysis is done what accuracy is needed. ‘When’ is when is it applied in the design process? Less is known at early design phases and therefore less accurate models are needed. Other useful definitions are:

Verification is the process of assuring that the equations are solved correctly. Numerical results are compared with known solutions. Verification is not discussed in this text. There exist several benchmark cases for checking finite element codes. A user should be aware that some unusual combinations may trigger problems that have not checked for by the code developer and no code is ever free of programming errors. *Validation* is when it is assured that the correct equations are solved. The analysis results are compared with reality.

Qualification is when it is assured that the conceptual model is relevant for the physical problem. The idealisation should be as large as possible – but not larger.

Sufficient valid and accurate solution is what the modelling process should result in. ‘Sufficient’ denotes that it must be related to the context the model is used in. For example, how accurate is loading known. It is no use to refine the model more than what is known about the real life problem. Then more must be found out about loading, material properties etc before improving the model.

Prediction is the final phase of where a simulation or analysis of a specific case that is different from validated case is done.

Uncertainty is of two types in the current context. Those can be removed by further investigations and those that cannot. This is related to variability which here denotes the variations that cannot be removed. This may be, for example, variation of material properties for different batches of nominally the same material or fluctuations of loading.

Simulation – an imitation of the internal processes and not merely the results of the system being simulated. This word is less precisely used in this text. Here the word is usually used when computing the evolution of a problem during a time interval. The word analysis is sometimes used to denote the computation of results at one instant of time.

3 Short finite element theory

The Finite Element Method is a numerical method for the approximate solution of most problems that can be formulated as a system of partial differential equations. There exist variants of the steps below that are needed in some cases. For the basic theory of the finite element is given in [2] and its application for nonlinear mechanical problems in [3]. The finite element method belongs to the family of weighted residual methods. There are two basic routes for formulating finite elements. The first one given below is general applicable whereas the second alternative requires the existence of a minimisation principle.

The difference between the approaches is in step 3 in the sections below. The Ritz method needs fewer manipulations at this step. The relation between Ritz method and FEM is also discussed in section 4.5.

3.1 Galerkin formulation of finite element methods

The Galerkin method is the basic approach used in this book as it is general applicable. It is in the same as variational methods and, in mechanics, the principle of virtual power. It is more a question of notations and naming (and history) that makes them seem different.

A short version of the basic steps can be described as below.

1. Make a guess, \hat{u} , (trial function) that has a number of unknown parameters. This is written as, for a problem with an unknown scalar field $u(x)$,

$$\hat{u}(x) = \sum_{i=1}^N u_i \varphi_i(x), \quad (3.1)$$

where the functions $\varphi_i(x)$ often are polynomials. The trial functions are set up in such a way that the N unknown coefficients (parameters) u_i are the field value at some point (node). This is, in the case of the standard finite element approach for mechanical problems, the nodal displacements u_i . It will be nodal temperature in the case of thermal problems.

2. The trial function is inserted into the partial differential equation and the boundary conditions of the problem. One variant of finite element formulation assumes that the essential boundary conditions are fulfilled. The remaining equations will not be fulfilled. Small errors – residuals are obtained. This will in our fictive case be

$$L(\hat{u}) = R \neq 0 \text{ and} \\ B_e(\hat{u}) = 0, B_n(\hat{u}) = R_n \neq 0. \quad (3.2)$$

L denotes the differential equation of the problem and is defined over the domain of the problem. B_e denotes the essential boundary conditions that are fulfilled and B_n are the natural boundary conditions that will be approximated. There exist other approaches for what equations and boundary conditions are fulfilled and not fulfilled but above is the most common one.

3. Make a weighted average of the errors to be zero. Therefore the name weighted residual method (WRM). Use the same functions as the trial functions as weighting functions. This variant of WRM makes it a Galerkin method. This step generates the same number of equations as number of unknowns.

$$\int_{\Omega} \varphi_k R dv + \int_{B_n} \varphi_k R_n dv = 0 \text{ for } k = 1, \dots, N. \quad (3.3)$$

4. Some manipulations leads to a system of coupled equations for the unknown parameters now in the array \mathbf{U} . This is the approximate solution.

$$\mathbf{KU} = \mathbf{F}_{ext}, \quad (3.4)$$

where \mathbf{K} is called a stiffness matrix in mechanical problems and \mathbf{F}_{ext} is a load vector due to different kind of loadings including possible natural boundary conditions. It is called heat conductivity matrix for thermal problems and the load vector is the heat input.

5. The method has theorems that promise convergence provided some conditions are fulfilled by the formulation. Thus an improved guess with more parameters will give a more accurate solution.
6. Derived quantities are often based on derivatives of the assumed solution in Eq. (3.1). They have larger errors than the primary variables in \mathbf{U} . The derivatives corresponds to strain and stress in mechanical problems and heat flux in thermal problems.

3.2 Ritz formulation of finite element methods

Many textbooks formulate the finite element method for mechanical, elastic problems using the theorem of minimum potential energy. This theorem requires that the essential boundary conditions of the problem must be fulfilled like we introduced in step 3 above. Which of the boundary conditions that are essential depends on what differential equation is solved. The steps in the Ritz approach are given below.

1. Make a guess (trial function) where a number of unknown parameters, this is the same as for the WRM approach,

$$\hat{u}(x) = \sum_{i=1}^N u_i \varphi_i(x). \quad (3.5)$$

2. The trial function is set into the functional² that should be stationary, most often minimum, for the exact solution. This functional is the total potential energy in mechanical problems. It is integrated over the domain like the first term in Eq. (3.3). Carrying out the integration leads to

$$\Pi(\mathbf{U}) = \frac{1}{2} \mathbf{U}^T \mathbf{KU} - \mathbf{U}^T \mathbf{F}_{ext}. \quad (3.6)$$

3. The condition that Eq. (3.6) is stationary w.r.t. \mathbf{U} leads to

$$\mathbf{KU} = \mathbf{F}_{ext}. \quad (3.7)$$

4. The method has theorems that promise convergence. Thus an improved guess with more parameters will give a more accurate solution.

6. Derived quantities are often based on derivatives of the assumed solution in Eq. (3.5). They have larger errors than the primary variables in \mathbf{U} . The derivatives corresponds to strain and stress in mechanical problems and heat flux in thermal problems.

3.3 Discretisation in space

Eqs (3.1) and (3.5) used nodal values as unknowns multiplying *trial functions*. The latter are usually defined over local regions, *elements*. Sometimes the trial functions are called *interpolations functions* as the field is interpolated between the nodal values using these functions. The interpolation within one single element is written as

$$\mathbf{u}_e(x) = \sum_{i=1}^{nnode} \mathbf{N}_i(x) \mathbf{u}_i = \mathbf{N} \mathbf{u}, \quad (3.8)$$

where \mathbf{N} is a matrix with interpolation 'functions, also often called *shape functions* as they determine the shape of the possible vector field on element can describe. \mathbf{u} is a vector with the

² [http://en.wikipedia.org/wiki/Functional_\(mathematics\)](http://en.wikipedia.org/wiki/Functional_(mathematics))

nodal values of all the nodes of an element. $nnode$ is the number of nodes in this element. The analysed geometry is split into elements. The elements are connected at the nodes as shown in Figure 3.1. The approximated field is interpolated over the elements from the nodal values. The elements with their shape functions must be combined so that there is no mismatch between the displacement fields along common boundaries of elements. There are certain continuity requirements that must be fulfilled when the trial function in Eq.s (3.1) or (3.5) are split into elementwise descriptions as in Eq. (3.8). The most crucial step in the finite element modelling process is the choice of elements and the discretisation of the domain. Refining the mesh and performing additional finite element solutions makes it possible to estimate the discretisation error.

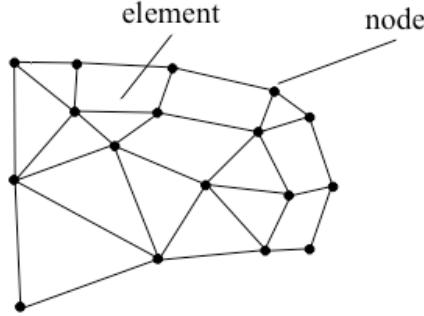


Figure 3.1. Discretised domain consisting of three and four node elements.

3.4 Tonti diagrams as a mean to illustrate relations

Different physics have different mathematical formulations but share some basic features. They are illustrated in Figure 3.2. The left side of the diagram are the kinematic variables, \mathbf{u} . The dynamic variables, \mathbf{f} , can be considered as the driving forces. However, it is perfectly possible to think reversely; that the kinematics are drivers and corresponding dynamic variables are results. The gradients, \mathbf{g} , are usually first or second derivatives of \mathbf{u} . The relation between the gradients and the flux depends on the medium and is therefore called material or constitutive relations. The fluxes are related to dynamic variables via some conservation or balance laws. There must be the same number of equations and unknowns. It is common to eliminate the fluxes and gradients and write a relation directly between \mathbf{u} and \mathbf{f} as denoted by the dashed lines in Figure 3.2.

The diagram can be used to describe the heat conduction equation, see Figure 3.3,

$$\frac{\partial}{\partial x} \left(\lambda_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda_z \frac{\partial T}{\partial z} \right) + Q = \rho c \frac{\partial T}{\partial t} . \quad (3.9)$$

The constitutive relation between temperature gradient and heat flux is called Fourier's law. It is written so that it allows different conductivities in different directions. They are usually the same. They can be temperature dependent and then they vary in space also. If not, then we can write the linear heat conduction equation as

$$\lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + Q = \rho c \frac{\partial T}{\partial t} . \quad (3.10)$$

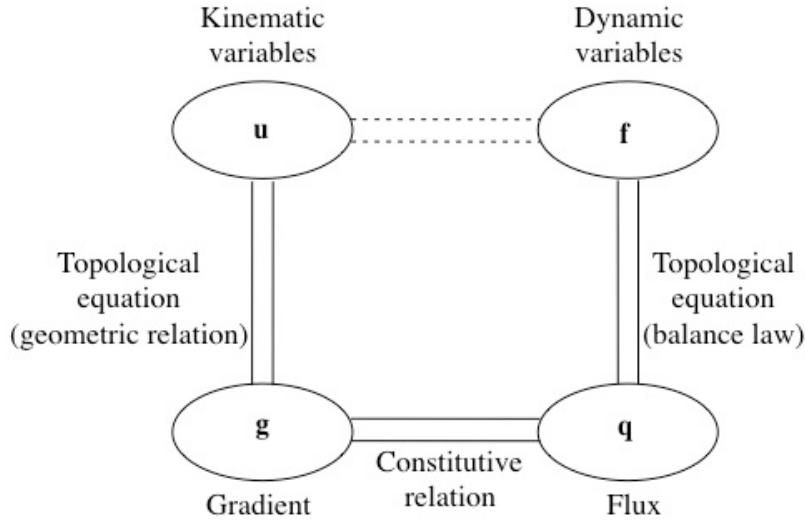


Figure 3.2. A Tonti diagram illustrating the basic variables and relations in a physical problem.

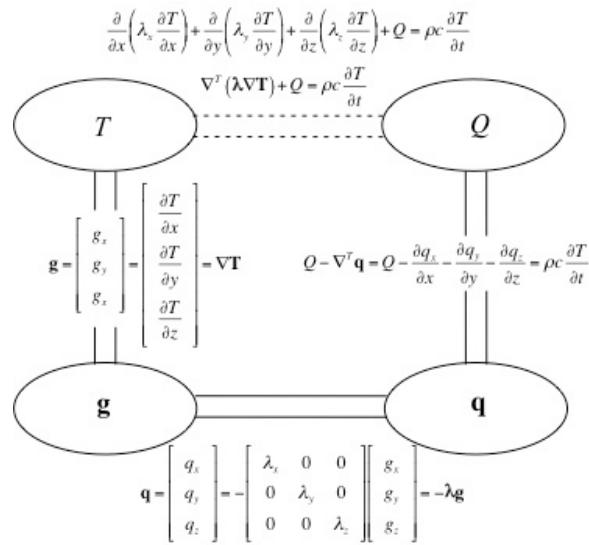


Figure 3.3. A Tonti diagram illustrating the basic variables and relations in transient heat conduction.

The diagram can also be used to illustrate mechanical problems³. Figure 3.4 shows the relations for static deformations of an elastic body. The constitutive equation is Hooke's law in matrix form, \mathbf{E} . It relates strain to stresses. The left side shows the displacement-strain relations and the right side expresses the equilibrium equations between volumetric loads and stresses. Other loads enter via boundary conditions specific for each problem. We will primarily study one-dimensional problems in the following. The actual appearance of the matrices in the figure depends on the problem type but also whether so-called tensor or vector notation is used for the strains and stresses, see section 15.1 in appendix. The Tonti diagram specific for this case is shown in Figure 3.5. E is here Young's modulus and the stresses and strains consist only of only one, axial, component. The equation

$$E \frac{d^2u}{dx^2} + F_x = \rho c \frac{d^2u}{dt^2} \quad (3.11)$$

³ http://en.wikipedia.org/wiki/Linear_elasticity

is the equation of motion expressed in terms of displacements as strains and stresses have been eliminated by means of the displacement-strain and constitutive relations. Eq. (3.11) is called Navier's equation.

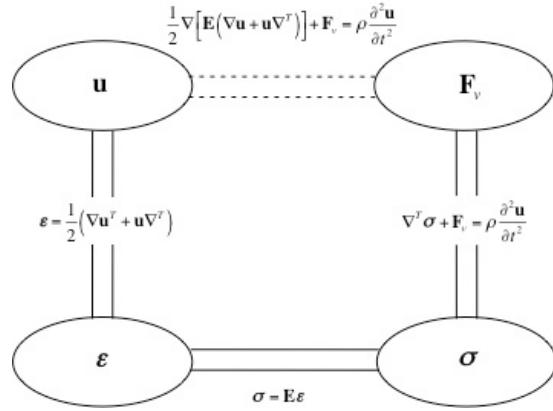


Figure 3.4. A Tonti diagram illustrating the basic variables and relations in dynamic deformation of elastic material.

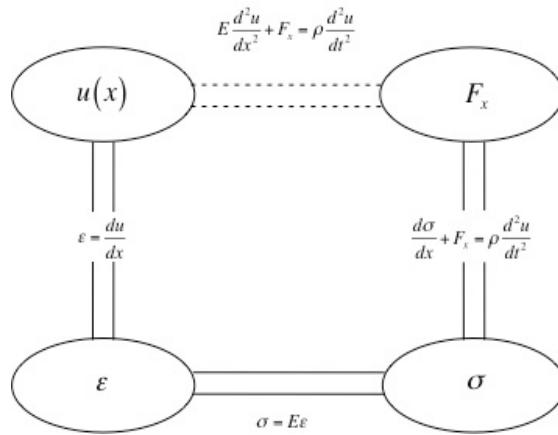


Figure 3.5. A Tonti diagram illustrating the equations for dynamic axial deformation of a rod.

The Tonti diagram can also be used to express the finite element formulation for a given problem. This is illustrated for the static deformation of a rod in Figure 3.6. The displacement field is interpolated over an element by

$$u^e(x) = \mathbf{N}(x)\mathbf{u} \quad (3.12)$$

The derivatives in the displacement-strain relation changes the matrix $\mathbf{N}(x)$ to \mathbf{B} giving

$$\epsilon = \frac{du^e}{dx} = \frac{d\mathbf{N}}{dx} \mathbf{u} = \mathbf{B}\mathbf{u} \quad (3.13)$$

in an element. Note that \mathbf{u} is the nodal displacements and does not depend on the coordinate. The material relation is the same as in Figure 3.5. Thus the relations from displacement to strain and stress is directly obtained by inserting $\mathbf{N}(x)\mathbf{u}$ into the fundamental equations in Figure 3.5. However, the balance equation between stresses external forces is now an integral. The reason for this will be clarified later. However, it indicated already here that the equilibrium statement for each point in a rod is replaced by an integral that will approximate equilibrium. It will be nodal equilibrium that can be written as

$$\mathbf{f}_{int} = \mathbf{f}_{ext} \quad (3.14)$$

The diagram is for one element and we will later reserve small letters for element variables and capital letters for corresponding relations for a whole structure (finite element model). The latter is written as

$$\mathbf{F}_{int} = \mathbf{F}_{ext}. \quad (3.15)$$

Thus the system of equations to be solved for a structure, Eq. (3.15), corresponds to nodal equilibrium. Investigating equilibrium of the finite element solution at other points or neighbourhoods may show that they are not in equilibrium.

The nodal equilibrium equation, Eq. (3.14), can be rewritten for this case of linear elastic deformation as

$$\int_{l^e} \mathbf{B}^T \sigma A dx = \int_{l^e} \mathbf{B}^T E \mathbf{B} A dx \mathbf{u} = \mathbf{k} \mathbf{u} = \mathbf{f}_{ext}. \quad (3.16)$$

\mathbf{k} is the element stiffness matrix is

$$\mathbf{k} = \int_{l^e} \mathbf{B}^T E \mathbf{B} A dx. \quad (3.17)$$

It will be shown later that this will lead to an expression for the whole structure written as
The equation is only fulfilled at the nodes with nodal equilibrium for the system written as

$$\mathbf{KU} = \mathbf{F}_{ext}. \quad (3.18)$$

Relation (3.15) is a more general form than in Eq. (3.18). It can be adapted to nonlinear problem also. The internal and external forces come from the weighted residual of the equilibrium equations in Eq. (3.3) with

$$R = E \frac{d^2 u}{dx^2} + F_x. \quad (3.19)$$

This will be detailed in chapter 6. Thus determining the solution by setting weighted residuals to zero can for mechanical problems be interpreted as approximating equilibrium everywhere to nodal at nodes.

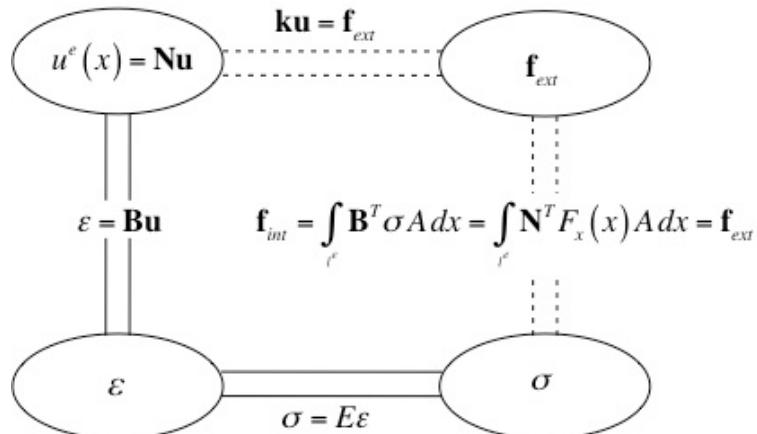


Figure 3.6. A Tonti diagram illustrating the finite element formulation for the axial deformation of a rod.

4 Weighted Residual Methods

The Weighted Residual Method is illustrated on the solution for a scalar field $u(x)$ in a simple one-dimensional geometry. First the problem is given a general mathematical form that is relevant for any differential equation. It is assumed that a problem is governed by the differential equation

$$L(u) = 0 . \quad (4.1)$$

It is to be solved over a given domain. The solution is subject to the initial conditions

$$I(u) = 0 \quad (4.2)$$

and boundary conditions

$$B(u) = 0 . \quad (4.3)$$

L , I and B denote operators on u . This can be derivatives and any kind of operations so they represent all kinds of mathematical problems. An approximate solution \hat{u} is inserted in to these relations giving residuals, errors,

$$L(\hat{u}) = R \neq 0 , \quad (4.4)$$

$$I(\hat{u}) = R_I \neq 0 \text{ and} \quad (4.5)$$

$$B(\hat{u}) = R_B \neq 0 . \quad (4.6)$$

The approximate solution can be structured so that;

- i) $R = 0$. Then it is called a boundary method.
- ii) $R_B = 0$. Then it is called an interior method. This requirement may be violated for some boundary conditions reducing the efficiency⁴ of the method.
- iii) Else it is a mixed method.

Boundary element methods⁵ are example of boundary methods. Green functions⁶ can be used as trial functions. Then only the boundary of the domain need be discretised. This will result in small, but full, matrices when the surface-volume ratio is small.

We focus on interior in the following examples of weighted residual methods. The approximate solution is taken as

$$\hat{u}(x) = u_0(x,t) + \sum_{i=1}^N a_i(t) \varphi_i(x), \quad (4.7)$$

where $\varphi_i(x)$ are analytic functions called the trial functions. $u_0(x,t)$ must satisfy initial and boundary conditions. However, we will later see cases where the trial functions are used for this purpose also and no separate $u_0(x,t)$ is used. The trial functions should be linearly independent and the first N members of the chosen set should be used. Notice that the parameters to be determined, $a_i(t)$, are chosen to be a function of time and the trial functions is only dependent on space. This is the most common approach although not necessary. One exception is space-time finite elements.

⁴ Efficiency in terms of needed number of terms to obtain a given accuracy. However, other advantages may be gained motivating a relaxing this requirement as will be shown later.

⁵ <http://www.boundary-element-method.com/intro.htm>

⁶ http://en.wikipedia.org/wiki/Green%27s_function

The parameters $a_i(t)$ are determined by setting the weighted average of the residual over the computational domain to zero

$$\int_{\Omega} w_k(x) R dv = 0 \quad \text{for } k = 1, \dots, N. \quad (4.8)$$

Additional terms may be included if the requirement on fulfilling all the boundary conditions is relaxed. The functions $w_k(x)$ are called weight functions. N independent equations are needed to determine the coefficients. Therefore, N independent weight functions are needed. If $N \rightarrow \infty$, then the residual will become zero in the mean, provided the initial and boundary conditions were fulfilled exactly, and thereby the approximate solution will converge to the exact solution in the mean

$$\lim_{N \rightarrow \infty} \int_{\Omega} \|\hat{u} - u_{exact}\|_2 dv = 0. \quad (4.9)$$

We will illustrate some variants of WRM methods in the following.

4.1 Subdomain method

The domain is split into subdomains, D_k , which may overlap. The weight function is then

$$w_k = \begin{cases} 1 & \text{in } D_k \\ 0 & \text{else} \end{cases} \quad \text{for } k = 1, \dots, N. \quad (4.10)$$

One example of this is the finite-volume method⁷. Each element is surrounding its associated node. Conservation equations then relate changes within this volume with fluxes over its boundaries.

4.2 Collocation method

The weight function is given by

$$w_k = \delta(x - x_k) \quad \text{for } k = 1, \dots, N, \quad (4.11)$$

where δ is the Dirac delta function⁸. Thus the residual is forced to be zero at specific locations.

4.3 Least-squares method

The weight function is given by

$$w_k = \frac{\partial R}{\partial a_k} \quad \text{for } k = 1, \dots, N, \quad (4.12)$$

where a_k are the coefficients in the approximate solution, Eq. (4.7). This makes the Eq. (4.8) corresponding to

$$\int_{\Omega} \frac{\partial R}{\partial a_k} R dv = 0 \quad \text{for } k = 1, \dots, N. \quad (4.13)$$

This in turn is the stationary value of

$$\Pi(a_1, a_2, \dots, a_N) = \int_{\Omega} R^2 dv, \quad (4.14)$$

thereby motivating the name of the method.

⁷ http://en.wikipedia.org/wiki/Finite_volume_method

⁸ http://en.wikipedia.org/wiki/Dirac_delta_function

4.4 Method of moments

The weight function is in this case given by

$$w_k = x^k \quad \text{for } k = 1, \dots, N-1. \quad (4.15)$$

4.5 Galerkin method

The weight function is chosen from the same family of functions as the trial functions in Eq. (4.7).

$$w_k = \varphi_k(x) \quad \text{for } k = 1, \dots, N. \quad (4.16)$$

The trial (or test) functions are taken from the first N members of a complete set of functions in order to guarantee convergence when increasing N .

The Galerkin method⁹ is the same as the *principle of virtual work or power*¹⁰ used in mechanics when formulating the finite element method. The weight functions correspond to virtual displacements or velocities in this approach. For elastic problems, it also corresponds to the *principle of minimum total energy*^{11,12}. This method can be a starting point for formulating approximate solutions. It is sometimes called Ritz method. It is commonly used in basic courses about the finite element method in mechanics. Then the essential boundary conditions must be fulfilled but the natural boundary conditions are approximated and included in the residual as shown in the next section.

However, Ritz method is not valid in cases like plasticity. Then the principle of virtual power is used in finite element textbooks for students in mechanics. These methods are also related to calculus of variations¹³.

The relation between the Galerkin and Ritz approaches for deriving the finite element method is also discussed in appendix, section 15.2.

4.6 Galerkin method including natural boundary condition in residual

We add the term

$$\int_{\Omega_n}^b w_k(x) R_n dv - \int_{\Omega_n} \varphi_k(x) R_n dv = 0 \quad \text{for } k=1, \dots, N \quad (4.17)$$

to Eq. (4.8). This means that we use the same weight functions over the boundary where we have the natural boundary conditions as we use in the interior but with opposite sign. The reason for this is that this will make some terms later cancel each other.

4.7 Petrov-Galerkin method

The weight function is represented by

$$w_k = P_k(x) \quad \text{for } k = 1, \dots, N-1, \quad (4.18)$$

where P_k is functions similar to the test functions φ_k but with additional terms to impose some additional requirements on the solution. Typically, terms to improve the solution of problems with convection like in convection-dominated fluid flow problems.

⁹ http://en.wikipedia.org/wiki/Galerkin_method

¹⁰ http://en.wikipedia.org/wiki/Virtual_work

¹¹ http://en.wikiversity.org/wiki/Introduction_to_Elasticity/Principle_of_minimum_potential_energy

¹² http://en.wikipedia.org/wiki/Minimum_total_potential_energy_principle

¹³ http://en.wikipedia.org/wiki/Euler-Lagrange_equation and http://en.wikipedia.org/wiki/Calculus_of_variations

4.8 Comparison of WRM methods

4.8.1 Problem definition and exact solution

The different weighted residual methods will be applied on the problem

$$L(u) = a \frac{d^2u}{dx^2} + bu - f_{ext} = A(u) - f_{ext} = 0 \quad \text{where } x \in [0,1]. \quad (4.19)$$

We will limit our discussion to the particular case of $a=1$ [m^2] and $b=1$ [m] and $f_{ext} = 1$. It is assumed that the function u^{14} is written in non-dimensional form, i.e. it does not have any dimensions.

$$B_e(0) = u(0) - 1 = 0 \quad (4.20)$$

and

$$B_n(0) = \left. \frac{du}{dx} \right|_{x=1} - \frac{1}{\cos(1)} = 0 \quad \text{or} \quad B_n(1) = \left. \frac{du}{dx} \right|_{x=1} - 1 = 0. \quad (4.21)$$

The constants have units that will not be visible in later discussions and thereby it may seem that the units are not consistent between different terms – but they are!

The boundary conditions have been named e =essential and n =natural for reasons shown later in the finite element formulation, chapter 6. The exact solution is the same for both boundary conditions but the first variant of Eq. (4.21) is easier to implement as it gives directly a condition for the value of one coefficient. The other variant gives a relation between the unknown coefficients that can be implemented in different ways. The exact solution to Eq.s (4.19)-(4.21) is

$$u_{exact} = \frac{\sin(x)}{\cos(1)} + 1. \quad (4.22)$$

4.8.2 Trial function and residuals

We choose an approximate solution given by

$$\hat{u} = \sum_{j=1}^N a_j \varphi_j = \sum_{j=1}^N a_j x^{j-1}. \quad (4.23)$$

This gives the residuals

$$L(\hat{u}) = \frac{d^2\hat{u}}{dx^2} + \hat{u} - 1 = \sum_{j=3}^N a_j (j-1)(j-2)x^{j-3} + \sum_{j=1}^N a_j x^{j-1} - 1 = R, \quad (4.24)$$

$$B_e(\hat{u}) = \hat{u}(0) - 1 = R_e \quad \text{and} \quad (4.25)$$

$$B_n(\hat{u}) = \hat{u}'(0) - \frac{1}{\cos(1)} = R_n \quad \text{or} \quad B_n(\hat{u}) = \hat{u}'(1) - 1 = R_n. \quad (4.26)$$

Fulfilling the essential boundary condition does not require the extra term u_0 in Eq. (4.7) but is achieved by setting

$$\hat{u}(0) = a_1 = 1. \quad (4.27)$$

The natural boundary condition can also be used to impose conditions on the parameters. Eq. (4.21) gives directly

¹⁴ The general solution is $u = C_1 \cos(x) + C_2 \sin(x) + 1$

$$\left. \frac{d\hat{u}}{dx} \right|_{x=0} = a_2 = \frac{1}{\cos(1)}. \quad (4.28)$$

The expression is somewhat more complex if imposed at $x=1$. Then we have

$$\left. \frac{d\hat{u}}{dx} \right|_{x=1} = \sum_{j=1}^N (j-1)a_j = \sum_{j=2}^N (j-1)a_j = 1. \quad (4.29)$$

This equation is called a constraint and can be used to remove one of the coefficients from the unknowns. If a_2 is removed this way, as a_1 is already fixed by the essential boundary conditions, then we write

$$a_2 = 1 - \sum_{j=3}^N (j-1)a_j. \quad (4.30)$$

Imposing this constraint can be done by the logic in appendix, section 15.3.

4.8.3 Subdomain weight functions

The *subdomain method*, section 4.1, splits the unit interval into N domains. We make them equal sized, Δx , and thus Eq. (4.8) becomes

$$\int_{x_{k-1}}^{x_k} R dv = 0 \text{ for } k = 1, \dots, N, \quad (4.31)$$

where

$$x_k = \frac{k}{N} = k\Delta x \text{ for } k = 1, \dots, N. \quad (4.32)$$

This can be integrated giving, for each k ,

$$\sum_{j=3}^N a_j (j-1) (x_k^{j-2} - x_{k-1}^{j-2}) + \sum_{j=1}^N \frac{a_j}{j} (x_k^j - x_{k-1}^j) = \Delta x. \quad (4.33)$$

This leads to a system of equations with

$$K_{kj} a_j = F_{ext,k}^{15}, \quad (4.34)$$

where the right hand side is

$$F_{ext,k} = 1/N \text{ for all } k \quad (4.35)$$

and the matrix on the left hand side becomes

$$K_{kj} = \frac{x_k^j - x_{k-1}^j}{j} \text{ and if } j > 2, (j-1)(x_k^{j-2} - x_{k-1}^{j-2}) \text{ is added.} \quad (4.36)$$

Table 4.1 below shows the results for the subdomain method and its condition number when applying the natural boundary condition at the left end. The error is defined by Eq. (4.9).

Table 4.2 shows the results if the natural boundary condition is imposed at the right end of the domain. This variant has a higher condition numbers causing problems when N increases. A high condition number and indicates that the solution is sensitive to round off error. It may even be impossible to invert the matrix.

Figure 4.1 below shows why the error for this particular case) is larger using the natural boundary conditions at the right end. The approximate solution enforces the wanted slope but it is then much lower than the theoretical solution.

¹⁵ This has same meaning as $\mathbf{Ka} = \mathbf{F}_{ext}$ where it is implied a summation over the repeated index j .

The implementation into Matlab™ is shown in Table 4.3 and Table 4.4.

Table 4.1. L2-error and condition number of matrix for different subdomain solutions using natural boundary condition at left end, Eq. (4.28).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.021	1.0000	9	1.2e-6	7.7e5
4	0.0035	27.8	10	3.3e-8	5.3e6
5	0.0008	284	11	2.3e-9	4.0e7
6	2.5e-3	2.0e3	12	1.4e-10	2.9e8
7	3.2e-4	1.5e4	13	2.3e-9	2.2e9
8	1.3e-5	1.0e5	14	6.9e-9	1.6e10

Table 4.2. L2-error and condition number of matrix for different subdomain solutions using natural boundary condition at right end, Eq. (4.30).

Number of terms (N)	Error	Condition number
3	0.044	1.0000
4	0.056	46.1
5	0.049	1380
6	0.056	92500
7	0.051	1.88e6
8	0.056	1.35e8

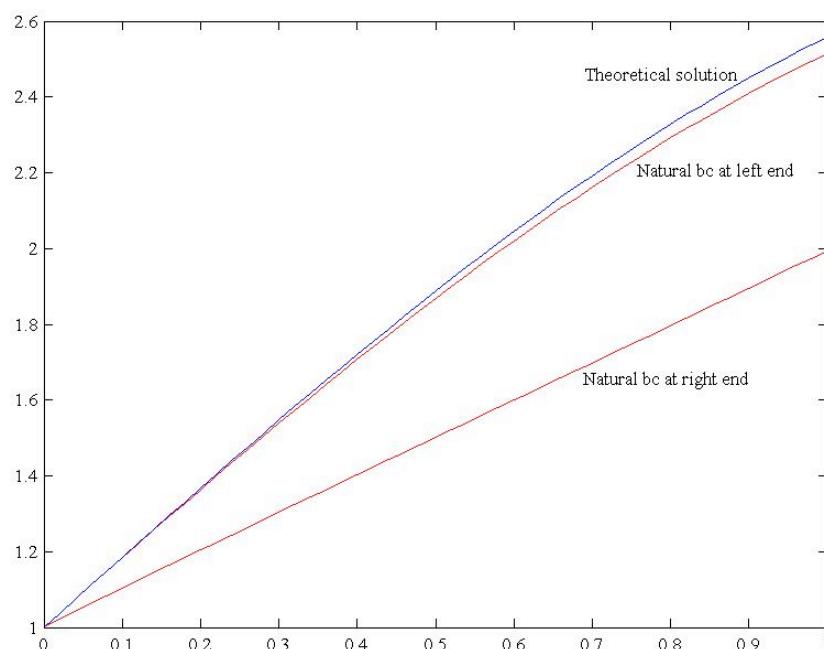


Figure 4.1. Plot of approximate solutions using four terms.

Table 4.3. Matlab™ code for assembling matrix and load vector for the subdomain method.

```

for k=1:N
    Fext(k)=dx;           % Eq. (4.35)
    xk=k*dx;
    xk_1=xk-dx;
    for j=1:N
        K(k,j)=(xk^j-xk_1^j)/j; % Eq. (4.36)
        if j>2
            K(k,j)=K(k,j)+(j-1)*(xk^(j-2)-xk_1^(j-2)); % Eq. (4.36)
        end
    end
end

```

Table 4.4. Matlab™ code for the subdomain method. The additional part for applying the boundary conditions at right end, Eq. (4.30), is shown. It is based on logic in appendix, section 15.3.

```

% We impose the condition a1=1 by multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% and move to right hand side as this is known
F=Fext-Fmod;
% The first equation for a1 is not needed any more - shrink the system
F(1)=[];
K(1,:)=[];
K(:,1)=[];
if alt==1
    %-----natural bc at left end
    a2=1/cos(1); Fmod=K(:,1)*a2;
    % Move this to right hand side
    F=F-Fmod;
    % The first equation for a2 is not needed - shrink the system
    F(1)=[]; K(1,:)=[]; K(:,1)=[];
    % NOTICE THAT NOW THE MATRIX IS JUST N-2XN-2
elseif alt==2
    %-----natural bc variant 2
    % We impose the bc on the second term a2=1-sum(j=3:N)aj(j-1)
    % (which now is first in our system of equations)
    % Thus A=[a2 a3 .. aN]' will be reduced to [a3 ..aN]
    % We can write A=Q*[a3 .. aN]'+[1 0 0 0 ..]'
    % See appendix, section 15.3.
    Q=zeros(N-1,N-2); C=zeros(N-1,1); C(1)=1;
    for j=3:N
        m=j-2;
        Q(1,m)=-(j-1);
        Q(m+1,m)=1;
    end
    % we will now reduce the problem further as we have
    % KA=F
    % and insert A=TAred+L
    % giving K(TAred+L)=F
    % KTAred=F-KL
    % and we premultiply by T' to make reduce the number of equations
    % T'*K*T*Ared=T'(F-KL)
    F=Q'**(F-K*L);
    K=T'*K*T;
    % NOTICE THAT NOW THE MATRIX IS JUST N-2XN-2
end

```

4.8.4 Collocation weight functions

The *collocation method*, section 4.2, requires the residual to be zero at specific locations. We make specify these points to be at centre of domains of equal length. Thus N locations cause these points to be at

$$x_k = \frac{k - 0.5}{N} = k\Delta x \text{ for } k = 1, \dots, N. \quad (4.37)$$

Then Eq. (4.8) becomes

$$\sum_{j=3}^N a_j (j-1)(j-2)x_k^{j-3} + \sum_{j=1}^N a_j x_k^{j-1} - 1 = 0 \text{ for } k = 1, \dots, N. \quad (4.38)$$

This leads to a system of equations with

$$K_{kj} a_j = F_{ext,k}, \quad (4.39)$$

where the right hand side is

$$F_k = 1 \text{ for all } k, \quad (4.40)$$

and the matrix on the left hand side becomes

$$K_{kj} = x_k^{j-1} \text{ and if } j > 2 + x_k^{j-3}. \quad (4.41)$$

The table below shows the results for the subdomain method and the implementation is shown in Table 4.6

Table 4.5. L2-error and condition number of matrix for collocation method using natural boundary condition at left end, Eq. (4.28).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.02	1.0	7	6.9e-6	1.5e4
4	0.0035	27.7	8	2.8e-7	1.0e5
5	0.0008	283.	9	2.6e-8	7.5e6
6	5.2e-5	2.e3	10	7.6e-10	5.3e7

Table 4.6. Matlab™ code for the point collocation method.

```

for k=1:N
    Fext(k)=1; % Eq. (4.40)
    xk=(k-0.5)*dx;
    for j=1:N
        K(k,j)=xk^(j-1); % Eq. (4.43)
        if j>2
            K(k,j)=K(k,j)+(j-2)*(j-1)*xk^(j-3); % Eq. (4.43)
        end
    end
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];

```

4.8.5 Weight functions in the least squares method

The *least squares method*, section 4.3, makes Eq. (4.8)

$$\int_0^1 \left((k-1)(k-2)x^{k-3} + x^{k-1} \right) \left[\sum_{j=3}^N a_j (j-1)(j-2)x^{j-3} + \sum_{j=1}^N a_j x^{j-1} - 1 \right] dx = 0 \quad (4.42)$$

where the first term is only present when $k > 2$. Further elaboration gives leads to a system of equations with

$$K_{kj} a_j = F_{ext,k}, \quad (4.43)$$

where the right hand side, coming from '-1'-term in R , is

$$F_{ext,k} = \frac{1}{k} \text{ and if } k > 2 \text{ } (k-1) \text{ is added.} \quad (4.44)$$

The matrix, that is symmetric, becomes

$$\begin{aligned}
K_{kj} &= \frac{1}{j+k-1} \\
\text{and if } k > 2 &\frac{(k-1)(k-2)}{j+k-3} \text{ is added,} \\
\text{and if } j > 2 &+ \frac{(j-1)(j-2)}{k+j-3} \\
\text{and if } k, j > 2 &+ \frac{(j-1)(j-2)(k-1)(k-2)}{k+j-5}.
\end{aligned} \quad (4.45)$$

The results are shown in Table 4.7 and the implementation in Table 4.8. It can be noted that the error increases at $N=10$. This is due to the high condition number. This trend goes on until $N=14$ where the solution procedure fails completely when using Matlab™.

Table 4.7. L2-error and condition number of matrix for least squares solution using natural boundary condition at left end, Eq. (4.28).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Reciprocal condition number
3	0.011	1.0000	7	1.3e-8	4.5e5
4	0.00018	37.4	8	2.0e-10	1.3e6
5	1.3e-5	736.	9	1.3e-11	4.0e8
6	2.1e-7	1.7e4	10	7.0e-11	1.1e10

Table 4.8. Matlab™ code for the least squares method.

```

for k=1:N
    k1k2=(k-1)*(k-2);
    Fext(k)=1/k;
    if k>2
        Fext(k)=Fext(k)+k-1;      % Eq. (4.50)
    end
    for j=1:N
        j1j2=(j-1)*(j-2);
        K(k,j)=1/(j+k-1);      % Eq. (4.51)
        if k>2 && j> 2
            K(k,j)=K(k,j)+(j1j2+k1k2)/(j+k-3)+j1j2*k1k2/(k+j-5);
            % Eq. (4.51)
        elseif k>2
            K(k,j)=K(k,j)+k1k2/(j+k-3); % Eq. (4.51)
        elseif j>2
            K(k,j)=K(k,j)+k1k2/(j+k-3); % Eq. (4.51)
        end
    end
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];K(1,:)=[];K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];K(1,:)=[];K(:,1)=[];

```

4.8.6 Method of moments weight functions

The *method of moments*, section 4.4, makes Eq. (4.8)

$$\int_0^1 x^k \left[\sum_{j=3}^N a_j (j-1)(j-2) x^{j-3} + \sum_{j=1}^N a_j x^{j-1} - 1 \right] dx = 0 \quad (4.46)$$

Further elaboration gives leads to a system of equations with

$$K_{kj} a_j = F_{ext,k} \quad (4.47)$$

where the right hand side, coming from '-1'-term in R , is

$$F_{ext,k} = \frac{1}{k+1} . \quad (4.48)$$

The matrix on the left hand side becomes

$$K_{kj} = \frac{1}{j+k} \text{ if } j > 2 + \frac{(j-1)(j-2)}{k+j-2}. \quad (4.49)$$

The results are shown in Table 4.9 and the implementation in Table 4.10. The error increases when $N=10$ and fails completely at $N=13$.

Table 4.9. L2-error and condition number of matrix for method of moments solution using natural boundary condition at left end, Eq. (4.28).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.02	1.0000	7	2.5e-6	2.9e6
4	0.0028	236.	8	7.6e-8	1.1e7
5	0.0005	1.7e4	9	6.5e-9	4.2e8
6	2.5e-5	8.0e5	10	5.2e-8	1.4e9

Table 4.10. Excerpt from Matlab™ code for method of moments.

```

for k=1:N
    Fext(k)=1/(k+1); % Eq. (4.48)
    for j=1:N
        K(k,j)=1/(j+k); % Eq. (4.50)
        if j>2
            K(k,j)=K(k,j)+(j-1)*(j-2)/(j+k-2); % Eq. (4.50)
        end
    end
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];

```

4.8.7 Galerkin weight functions

The *Galerkin method*, section 4.5, will be quite similar to the method of moments, section 4.4, for this particular choice of trial functions. Eq. (4.8) becomes

$$\int_0^1 x^{k-1} \left[\sum_{j=3}^N a_j (j-1)(j-2) x^{j-3} + \sum_{j=1}^N a_j x^{j-1} - 1 \right] dx = 0. \quad (4.50)$$

This leads to

$$\sum_{j=3}^N a_j \frac{(j-1)(j-2)}{k+j-3} 1^{k+j-3} + \sum_{j=1}^N \frac{a_j}{k+j-1} 1^{k+j-1} - \frac{1^k}{k} = 0. \quad (4.51)$$

Further elaboration gives leads to a system of equations with

$$K_{kj} a_j = F_{ext,k} \quad (4.52)$$

where the right hand side, coming from '-1'-term in R , is

$$F_{ext,k} = \frac{1}{k}. \quad (4.53)$$

The matrix on the left hand side becomes

$$K_{kj} = \frac{1}{j+k-1} \text{ and if } j > 2 + \frac{(j-1)(j-2)}{k+j-3}. \quad (4.54)$$

The results are shown in Table 4.11 and the implementation in Table 4.13. An implementation based on applying the boundary condition at $x=1$ is also used. The result for this case is shown in Table 4.12.

Table 4.11. L2-error and condition number of matrix for Galerkin method using natural boundary condition at left end, Eq. (4.28).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.019	1.0000	7	1.2e-6	9.8e6
4	0.002	145.	8	3.1e-8	3.3e7
5	0.00033	7.5e3	9	2.1e-9	1.2e10
6	1.3e-5	2.9e5	10	3.1e-9	3.8e11

Table 4.12. L2-error and condition number of matrix for Galerkin method using natural boundary condition at right end, Eq. (4.30).

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.0015	18.3	7	5.2e-9	8.5e6
4	4.2e-5	375.	8	1.0e-10	2.7e8
5	3.7e-6	1.0e4	9	2.2e-10	8.4e9
6	8.1e-8	2.8e5	10	4.8e-8	2.5e11

Table 4.13. Matlab™ code for Galerkin's method using natural boundary condition at left end, Eq. (4.28).

```

for k=1:N
    Fext(k)=1/k;      % Eq. (4.53)
    for j=1:N
        K(k,j)=1/(j+k-1); % Eq. (4.54)
        if j>2
            K(k,j)=K(k,j)+(j-1)*(j-2)/(j+k-3); % Eq. (4.54)
        end
    end
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];

```

4.8.8 Galerkin method including natural boundary condition in residual

The additional term to the residual in Eq. (4.51) comes from the natural boundary condition in Eq. (4.17). The natural boundary condition is applied at $x=1$ for the current example. This gives the additional term

$$-\int_{\Omega_n} x^{k-1} \left(\left[\frac{d}{dx} \sum_{j=1}^N a_j x^{j-1} \right]_{x=1} - 1 \right) dv \quad \text{for } k=1,\dots,N. \quad (4.55)$$

This leads to

$$-\left[x^{k-1} \left(\frac{d}{dx} \sum_{j=1}^N a_j x^{j-1} - 1 \right) \right]_{x=1} = \left[\sum_{j=1}^N 1 - (j-1)a_j \right]_{x=0} \quad \text{for } k=1,\dots,N. \quad (4.56)$$

The result is summarised in Table 4.14. Notice that for $N=3$ we have two unknowns whereas the previous variant has only one unknown.

Table 4.14. L2-error and condition number of matrix for Galerkin method including natural boundary condition at right end, Eq. (4.28), in the residual.

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.0014	18.4	7	5.2e-9	8.5e6
4	4.2e-5	375.	8	0.9e-10	0.27e9
5	3.6e-6	1.0e4	9	2.8e-11	8.3e9
6	0.8e-7	2.8e5	10	3.1e-8	2.5e11

4.8.9 Summary of comparisons

The above comparisons illustrate the difference between the methods but are too limited to draw general conclusions. The table below is from Fletcher [4] with subjective comparisons of some of the methods.

Table 4.15. Subjective comparisons of different weighted residual methods, from Fletcher [4].

MWR	Galerkin	Least-squares	Subdomain	Collocation
Accuracy	Very high	Very high	High	Moderate
Ease of formulation	Moderate	Poor	Good	Very good
Additional remarks	Equivalent to Ritz method where applicable.	Not suited to eigenvalue or evolutionary problems.	Equivalent to finite-volume method; suited to conservation formulation.	Orthogonal collocation gives high accuracy.

4.9 Exercises

Exercise 4.1

Consider in $x \in [0, 1]$, the second order Boundary Value Problem

$$\frac{d^2u}{dx^2} + u + x = 0,$$

subject to

$$u(0) = \frac{du}{dx} \Big|_{x=1} = 0.$$

It has the exact solution

$$u_{exact} = \frac{\sin(x)}{\cos(1)} - x .$$

Find an approximate solution using the Galerkin method. Assume a trial function of the form

a)

$$\hat{u} = a_1 \varphi_1 = a_1 x (1-x)$$

b)

$$\hat{u} = a_1 \varphi_1 = a_1 x$$

c)

$$\hat{u} = a_1 \varphi_1 + a_2 \varphi_2 = a_1 x + a_2 x^2$$

Notice that we have only one parameter and that the trial function already satisfies the essential boundary condition. The assumed solution in case a) would have worked much

better if we have had $u(1)=0$ instead of $\frac{du}{dx}\Big|_{x=1}=0$ as boundary condition.

$$\text{Answer: a) } a_1 = \frac{36}{10} \text{ b) } a_1 = \frac{4}{11} \text{ c) } \begin{bmatrix} \frac{2}{3} & \frac{3}{4} \\ \frac{3}{4} & \frac{17}{15} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{4} \end{bmatrix} \Rightarrow \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \approx \begin{bmatrix} 0.9856 \\ -0.4317 \end{bmatrix} .$$

Exercise 4.2

Consider in $x \in [0, 1]$, the second order Boundary Value Problem

$$\frac{d^2u}{dx^2} + \frac{1}{1+x} = 0$$

subject to

$$u(0) = u(1) = 0$$

It has the exact solution

$$u_{exact} = 2 \ln(2) \cdot x - (1+x) \ln(1+x) .$$

Find an approximate solution using the Galerkin method. Assume a trial function of the form

$$\hat{u} = \sum_{j=1}^N a_j x^j .$$

Notice that we skipped the first term a_0 as $a_0=0$ fulfils essential boundary conditions. Solve the problem for $N=2$.

Clue: You will need

$$\int \frac{x - x^2}{1+x} dx = 3(1+x) - 2 \ln(1+x) - \frac{(1+x)^2}{2} .$$

$$\text{Answer: } a_1 = -a_2 = \frac{9}{2} - 6 \ln(2) \approx 0.341 .$$

5 Classical and computational Galerkin methods

The classical Galerkin methods were applied before computers were available. Thus there was a need to obtain high accuracy with few unknowns. The method used global test functions. The use of orthogonal¹⁶ test functions further reduced the calculations needed. Naturally, the use of global functions made it difficult to solve problems with irregular boundaries. Furthermore, the advent of computers made it possible to solve problems with greatly increased number of parameters. Today the results system of equations can have millions of unknowns.

The global test functions in classical Galerkin methods become less and less unique with increasing number. For example, going from a polynomial of x^{19} to x^{20} does not add much new information into Eq. (4.7). Thus adding more terms in the approximate solution will make the contribution from higher order terms smaller and smaller for larger N . This will then cause the system of equations to be solved to become ill-conditioned, i.e. sensitive to round-off and truncation errors.

Therefore the trial functions in computational Galerkin methods are chosen in order to reduce this problem. The use of spectral methods reduces this problem due to the orthogonal property of these functions. Finite Element Methods, the focus of the current text, are based on the use of local trial functions instead. Increasing the number of domains, elements, is the way the number of coefficients is increased. The way these domains, elements, are formulated is also a key to the success of Finite Element Methods giving the ability to solve various kind of physical problems with complex boundaries. The codes consist of general processing parts and the physics of the problem at hand is embedded in the element formulation. These elements are set up so that they can be used to discretise any geometry.

5.1 Exercise

Exercise 5.1 (Computational problem)

Consider in $x \in [-L/2, L/2]$, the second order Boundary Value Problem

$$\lambda \frac{d^2u}{dx^2} + q = 0,$$

where $q=a+bx$. The solution is subject to the boundary conditions

$$u\left(-\frac{L}{2}\right) = 0 \text{ and } u\left(\frac{L}{2}\right) = 0$$

Find an approximate solution using a spectral method, i.e. in this case using cosine and sine terms that fulfils the essential boundary conditions at the edges. The use of functions makes this corresponds to determining an approximate solution using a Fourier series to represent the solution. Notice that most terms in the integral are orthogonal and you will get a diagonal matrix.

Clue: The cosine series will represent the even part of the solution due to the load term a whereas the sine series will represent the odd part of the solution due to the bx term

Implement the solution in Matlab and evaluate it with respect to the exact solution. Make the comparison both for u itself as well as its first derivative, $\frac{du}{dx}$. Perform a convergence study like, for example, in Table 4.14.

¹⁶ http://en.wikipedia.org/wiki/Orthogonal_functions $\int \varphi_k \varphi_j dV = 0 \text{ if } k \neq j$

6 Finite Element Methods

There exist not only a variety of finite elements for different physical problems but also different formulations. The most commonplace variant is in focus in the following, see Figure 6.1. The relation between flux \mathbf{q} and gradient \mathbf{g} is fulfilled exactly together with the geometric relations between \mathbf{g} and \mathbf{u} . The variables \mathbf{q} and \mathbf{g} can then be eliminated and the balance law is expressed directly as a relation between \mathbf{u} and \mathbf{f} . However, this balance law is not fulfilled exactly but gives a residual \mathbf{R} . A weighted error of this residual is set to zero leading to a system of equations for the unknown parameters. This is called the displacement based finite element formulation in mechanics. The balance law is then the equation of motion or static equilibrium. It is approximately fulfilled by a system of equations expressing nodal equilibrium. Consistent with this approach is to use an approximate solution that fulfils the essential boundary conditions that also are geometric relations but include the natural boundary conditions in the residual as they are balance conditions at the boundary.

A Finite Element Method (FEM) solution of the same problem as earlier will be formulated with the use of the Galerkin approach but now allowing the trial solution to approximate the natural boundary condition and including it into the residual as in section 4.6. Furthermore, partial integration will be used in order to create a symmetric formulation with the same derivatives on the trial and weight functions. Global functions and local functions, the latter is the FEM, will be used. Two approaches for implementing FEM will be shown. The first case is based on the use of nodal based trial functions whereas the second variant is based on element based definitions. The latter is the more effective and is the way it is done in all computer codes. Iso-parametric formulation, which is the key factor to solve problems for arbitrarily geometries, will be introduced in this context.

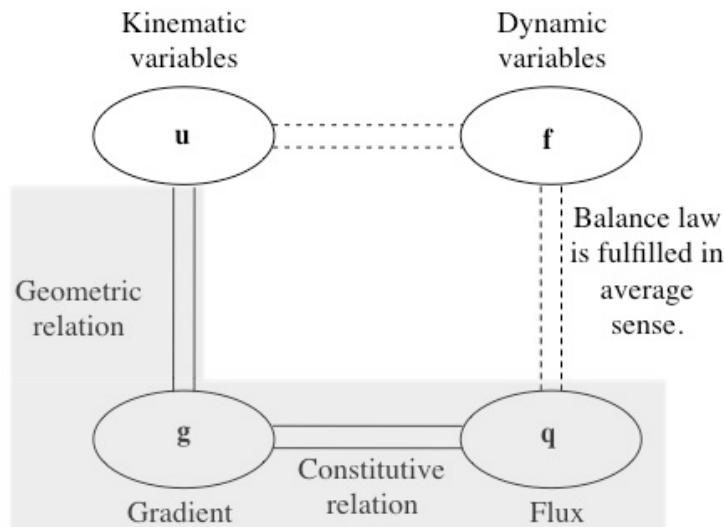


Figure 6.1. A Tonti diagram illustrating the most common approach for formulating the finite element method for a given problem. Grey area denotes relations fulfilled and used to eliminate \mathbf{g} and \mathbf{q} expressing balance law in \mathbf{u} .

The problem given in section 4.8 will be solved but the natural boundary condition will be prescribed at the right end and will be approximated in the solution¹⁷. The equations are repeated below.

$$L(u) = \frac{d^2u}{dx^2} + u - 1 = 0 \quad \text{where } x \in [0,1] \text{ with boundary conditions} \quad (6.1)$$

$$B_e(0) = u(0) - 1 = 0 \quad \text{and} \quad (6.2)$$

$$B_n(1) = \left. \frac{du}{dx} \right|_{x=1} - 1 = 0. \quad (6.3)$$

The exact solution to Eq.s (6.1)-(6.3) is still

$$u_{\text{exact}} = \frac{\sin(x)}{\cos(1)} + 1. \quad (6.4)$$

The natural boundary condition is now included in the residual by extending Eq. (4.8) with the error in R_B from Eq. (4.6) giving the same equation as Eq. (4.17). It is

$$\int_{\Omega} w_k(x) R dv - \int_{S_n} w_k(x) R_B ds = 0 \quad \text{for } k=1,\dots,N, \quad (6.5)$$

where S_n is the part of the boundary where the natural boundary conditions are prescribed (in this case $x=1$). We have chosen to use same weight functions for R_B . The reason for the choice of minus-sign will be obvious later as this will make some terms cancel each other.

Then we get

$$\int_{\Omega} w_k \left(\frac{d^2 \hat{u}}{dx^2} + \hat{u} - 1 \right) dv - \left[w_k \left(\frac{d\hat{u}}{dx} - 1 \right) \right]_{x=1} = 0 \quad \text{for } k=1,\dots,N. \quad (6.6)$$

There is a second derivative on the trial functions but no on the weight functions. We perform a partial integration¹⁸ leading to

$$\left[\frac{d\hat{u}}{dx} w_k \right]_0^1 - \int_{\Omega} \frac{dw_k}{dx} \frac{d\hat{u}}{dx} dv + \int_{\Omega} w_k (\hat{u} - 1) dv - \left[w_k \left(\frac{d\hat{u}}{dx} - 1 \right) \right]_{x=1} = 0 \quad \text{for } k=1,\dots,N. \quad (6.7)$$

The approximate solution must fulfil the essential boundary condition at $x=0$. Thus we can freely choose the weight function to be zero along this part of the boundary S_e or in our case $w_k(0) = 0$. Then we get

$$\left[\frac{d\hat{u}}{dx} w_k \right]_{x=1} - \int_{\Omega} \frac{dw_k}{dx} \frac{d\hat{u}}{dx} dv + \int_{\Omega} w_k (\hat{u} - 1) dv - \left[w_k \left(\frac{d\hat{u}}{dx} - 1 \right) \right]_{x=1} = 0 \quad \text{for } k=1,\dots,N. \quad (6.8)$$

Now we can see that the first term and the corresponding part of the last term cancel, as we introduced the minus sign in Eq. (6.5) giving

$$- \int_{\Omega} \frac{dw_k}{dx} \frac{d\hat{u}}{dx} dv + \int_{\Omega} w_k (\hat{u} - 1) dv - \left[w_k (-1) \right]_{x=1} = 0 \quad \text{for } k=1,\dots,N \quad (6.9)$$

or

¹⁷ We cannot approximate the natural boundary condition at $x=0$ as we will later set the weight function to zero due to the essential boundary condition also at $x=0$. This is no limitation and physics motivate, to only allow certain combinations of boundary conditions at the same boundary.

¹⁸ $\int f'g = [fg] - \int fg' \rightarrow \int u''w = [u'w] - \int u'w'$

$$\int_{\Omega} \frac{dw_k}{dx} \frac{d\hat{u}}{dx} dv - \int_{\Omega} w_k (\hat{u} - 1) dv = [w_k \cdot 1]_{x=1} \quad \text{for } k = 1, \dots, N. \quad (6.10)$$

If it is a mechanical problem, then the first term would be called a stiffness matrix. The second term would be nodal forces due to distributed loads and the term of the right hand side due to force on the boundary.

The formulation is now symmetric with respect to the weight and trial functions. This will lead to symmetric matrices giving computational efficiency. We will go through the details in the following using a global defined trial function and a locally defined as in FEM.

6.1 Global weight and trial functions

Eq. (6.10) is first used with the same global functions as used in section 4.6. We will in the next section introduce the finite element version with local functions. The approximate solution and corresponding weight functions are

$$\hat{u} = \sum_{j=1}^N a_j \phi_j = \sum_{j=1}^N a_j x^{j-1}. \quad (6.11)$$

Inserted into Eq. (6.10) gives

$$\int_0^1 \sum_{j=1}^N a_j (k-1)(j-1) x^{k+j-4} dx - \int_0^1 \left(\sum_{j=1}^N a_j x^{k+j-2} - x^{k-1} \right) dx = [x^{k-1} \cdot 1]_{x=1} \quad \text{for } k=1, \dots, N, \quad (6.12)$$

where the $k, j > 1$ for the first term. Notice the relaxed criterion on number of derivatives that must be possible to define, compared to Eq. (4.54) where j starts with 3 ($j > 2$) in the summation. This reduction is due to the partial integration performed. Evaluating the integrals gives

$$\left[\sum_{j=1}^N a_j \frac{(k-1)(j-1)}{k+j-3} x^{k+j-3} \right]_0^1 - \left[\sum_{j=1}^N \frac{a_j}{k+j-1} x^{k+j-1} - \frac{x^k}{k} \right]_0^1 = [x^{k-1} \cdot 1]_{x=1} \quad \text{for } k=1, \dots, N. \quad (6.13)$$

Finally we get

$$\sum_{j=1}^N a_j \frac{(k-1)(j-1)}{k+j-3} - \sum_{j=1}^N \frac{a_j}{k+j-1} = 1 - \frac{1}{k} \quad \text{for } k = 1, \dots, N. \quad (6.14)$$

A comparison with Eq. (4.52) shows that the problem is now symmetric with respect to indexes j and k . We have now

$$K_{kj} a_j = F_{ext,k} \quad (6.15)$$

with

$$F_{ext,k} = 1 - \frac{1}{k}. \quad (6.16)$$

The matrix on the left hand side becomes

$$K_{kj} = -\frac{1}{j+k-1} \text{ and if } j, k > 1 \text{ then } \frac{(k-1)(j-1)}{k+j-3} \text{ is added.} \quad (6.17)$$

Imposing the essential boundary condition is done like in earlier cases by prescribing $a_1=1$. The results shown are shown in Table 6.1 and the implementation in Table 6.2. It can be noted that the errors in Table 6.1 are the same as in Table 4.11. This is due to the fact that the partial integration only makes the matrix symmetric but will bring in any new assumptions compared to the derivation in section 4.8.8 and therefore gives identical results at lower cost. The error increases from $N=8$ and the method fails completely at $N=13$ due to the ill-condition system of equations.

Table 6.1. L2-error and condition number of matrix for Ritz method.

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.0015	18.3	7	5.2e-9	8.5e6
4	4.2e-5	375.	8	9.1e-11	2.7e8
5	3.7e-6	1.0e4	9	6.2e-10	8.4e9
6	8.1e-8	2.8e5	10	1.1e-8	2.5e11

Table 6.2. Matlab™ code for Ritz method case.

```

for k=1:N
    Fext(k)=1-1/k;      % Eq. (6.16)
    for j=1:N
        K(k,j)=-1/(j+k-1);    % Eq. (6.16)
        if j>1 && k >1
            K(k,j)=K(k,j)+(j-1)*(k-1)/(j+k-3);    % Eq. (6.16)
        end
    end
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];
K(1,:)=[];
K(:,1)=[];

```

6.2 Nodal based trial and weight functions

Now we use locally defined trial, and weight, functions of the type that are used in the finite element method. They are defined as, see Figure 6.2,

$$\varphi_j = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}} & \text{for } x_{j-1} \leq x \leq x_j \\ \frac{x_{j+1} - x}{x_{j+1} - x_j} & \text{for } x_j \leq x \leq x_{j+1} \end{cases}, \quad (6.18)$$

where

$$\Delta x = x_j - x_{j-1} = x_{j+1} - x_j \quad (6.19)$$

is the length of each segment and assumed constant for simplicity in the following. Note that the very first node ($j=1$) will only have the right half of the function and the final node ($j=N$) will only have the left part of the function.

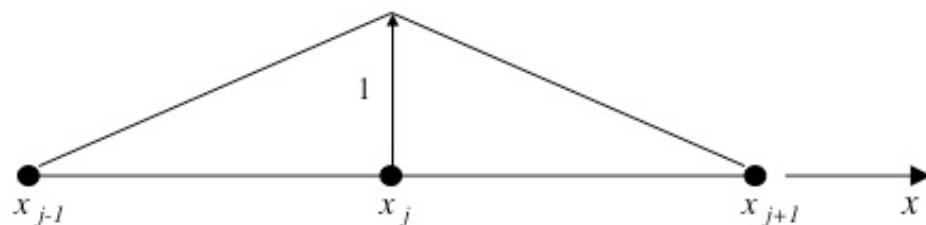


Figure 6.2. Finite element type of trial functions or shape functions with local base.

The approximate solution is then written as

$$\hat{u} = \sum_{j=1}^N u_j \varphi_j. \quad (6.20)$$

We change the notation now and use u_j instead of a_j . This is reasonable as we use functions that have the property

$$\varphi_j(x_i) = \delta_{ij}. \quad (6.21)$$

This means that it has unit value at the coordinate corresponding to $j=i$ and zero at all other nodal coordinates. The coefficients in Eq. (6.18) will therefore correspond to the nodal value at the specified node

$$\hat{u}(x_j) = u_j. \quad (6.22)$$

Furthermore, it has the partition of unity property¹⁹. The sum of the values of all functions overlapping at x is unit,

$$\sum_j \varphi_j(x) = 1. \quad (6.23)$$

Eq. (6.21) motivates the change in from a :s to u :s in the notations. Note also the introduction of the concept *node*. The trial functions φ_j are associated with a point at x_j that now will be called a node. For simplicity we have assumed that the distance between each node is equal in the example we are discussing. However, this is not necessary. FEM is very flexible with respect to geometry. This will be particularly clear later when discussing element based functions where the concept of an element is introduced. The discussion of isoparametric element formulation is the top of the line in the development of the finite element method giving its ultimate flexibility. Now – back on track!

Insertion of Eq. (6.18) into Eq. (6.10) gives

$$\int_0^1 \frac{d\phi_k}{dx} \sum_{j=1}^N u_j \frac{d\phi_j}{dx} dx - \int_0^1 \phi_k \left(\sum_{j=1}^N u_j \phi_j - 1 \right) dx = [\phi_k \cdot 1]_{x=1} \quad \text{for } k=1,\dots,N. \quad (6.24)$$

The integral can be split into separate contributions as the different functions only overlap in certain intervals. Thus for a given k , there is an overlap with right half of the function φ_j for $j=k-1$ and left half for $j=k+1$ according to the grey area in Figure 6.3.

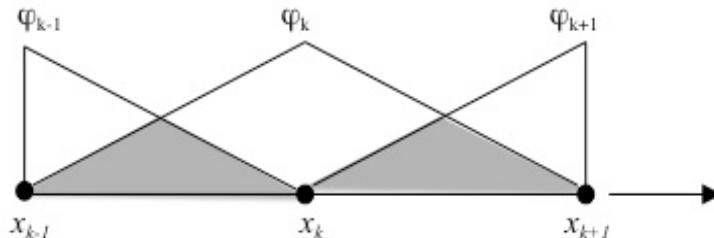


Figure 6.3. Overlap between neighbours where the trial functions contribute to integrals to be evaluated. Thus equation k will have contributions from three trial functions $j=k-1, j=k$ and $j=k+1$.

Then we can limit the evaluation of the integral for given k by

¹⁹ Called partition of unity, see http://en.wikipedia.org/wiki/Partition_of.Unity

$$\int_{x_{k-1}}^{x_{k+1}} \frac{d\varphi_k}{dx} \sum_{j=k-1}^{k+1} u_j \frac{d\varphi_j}{dx} dx - \int_{x_{k-1}}^{x_{k+1}} \varphi_k \left(\sum_{j=k-1}^{k+1} u_j \varphi_j - 1 \right) dx = 0 \text{ for } k = 2 \dots N-1. \quad (6.25)$$

Notice the exception for $k=1$ and $k=N$ as they are obtained from

$$\int_{x_1=0}^{x_2} \frac{d\varphi_k}{dx} \sum_{j=k}^{k+1} u_j \frac{d\varphi_j}{dx} dx - \int_{x_1=0}^{x_2} \varphi_k \left(\sum_{j=k}^{k+1} u_j \varphi_j - 1 \right) dx = 0 \text{ for } k = 1 \text{ and} \quad (6.26)$$

$$\int_{x_{N-1}}^{x_N=1} \frac{d\varphi_k}{dx} \sum_{j=N-1}^N u_j \frac{d\varphi_j}{dx} dx - \int_{x_{N-1}}^{x_N=1} \varphi_k \left(\sum_{j=N-1}^N u_j \varphi_j - 1 \right) dx = [\varphi_N \cdot 1]_{x=1} \text{ for } k = N. \quad (6.27)$$

We will make some observations before we formulate a more efficient variant of the finite element method than using the description in Figure 6.2. We will see that we integrate twice over the same region. First to give a contribution to a node on the left side of the segment and then for giving contribution to the node on the right side. This observation is the basis for the standard way to implement the finite element method based on the concepts of elements. An element corresponds to a region in which the unknown field is interpolated from connected nodes.

We will use three nodes for the Eq. (6.24) shown in the figure below. Eq.s (6.25)-(6.27) becomes

$$\int_{x_1=0}^{x_2} \frac{d\varphi_1}{dx} \sum_{j=1}^2 u_j \frac{d\varphi_j}{dx} dx - \int_{x_1=0}^{x_2} \varphi_1 \left(\sum_{j=1}^2 u_j \varphi_j - 1 \right) dx = 0 \text{ for } k=1, \quad (6.28)$$

and for $k=2$ we get

$$\int_{x_1}^{x_3} \frac{d\varphi_k}{dx} \sum_{j=1}^3 u_j \frac{d\varphi_j}{dx} dx - \int_{x_1}^{x_3} \varphi_k \left(\sum_{j=1}^3 u_j \varphi_j - 1 \right) dx = 0. \quad (6.29)$$

Finally for $k=N=3$ we get

$$\int_{x_2}^{x_3=1} \frac{d\varphi_k}{dx} \sum_{j=2}^3 u_j \frac{d\varphi_j}{dx} dx - \int_{x_2}^{x_3=1} \varphi_k \left(\sum_{j=2}^3 u_j \varphi_j - 1 \right) dx = [\varphi_3 \cdot 1]_{x=1}. \quad (6.30)$$

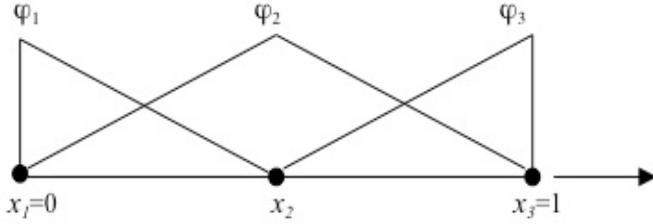


Figure 6.4. Three nodes used to solve Eq.s (6.1)-(6.3).

We can elaborate further to

$$\int_{x_1=0}^{x_2} \varphi'_1 [u_1 \varphi'_1 + u_2 \varphi'_2] dx - \int_{x_1=0}^{x_2} \varphi_1 (u_1 \varphi_1 + u_2 \varphi_2 - 1) dx = 0 \text{ for } k=1, \quad (6.31)$$

$$\int_{x_1}^{x_3} \varphi'_2 [u_1 \varphi'_1 + u_2 \varphi'_2 + u_3 \varphi'_3] dx - \int_{x_1}^{x_3} \varphi_2 (u_1 \varphi_1 + u_2 \varphi_2 + u_3 \varphi_3 - 1) dx = 0 \text{ for } k=2 \quad (6.32)$$

and

$$\int_{x_2}^{x_3=1} \varphi'_3 [u_2 \varphi'_2 + u_3 \varphi'_3] dx - \int_{x_2}^{x_3=1} \varphi_3 (u_2 \varphi_2 + u_3 \varphi_3 - 1) dx = [\varphi_3 \cdot 1]_{x=1} \text{ for } k=3. \quad (6.33)$$

The nodal variables do not depend on the coordinates and can be moved outside the integrals and we write

$$\int_{x_1=0}^{x_2} \varphi'_1 \varphi'_1 dx u_1 + \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_2 dx u_2 - \int_{x_1=0}^{x_2} \varphi_1 \varphi_1 dx u_1 - \int_{x_1=0}^{x_2} \varphi_1 \varphi_2 dx u_2 = - \int_{x_1=0}^{x_2} \varphi_1 dx \text{ for } k=1, \quad (6.34)$$

$$\begin{aligned} & \int_{x_1}^{x_3} \varphi'_2 \varphi'_1 dx u_1 + \int_{x_1}^{x_3} \varphi'_2 \varphi'_2 dx u_2 + \int_{x_1}^{x_3} \varphi'_2 \varphi'_3 dx u_3 - \\ & \int_{x_1}^{x_3} \varphi_2 \varphi_1 dx u_1 - \int_{x_1}^{x_3} \varphi_2 \varphi_2 dx u_2 - \int_{x_1}^{x_3} \varphi_2 \varphi_3 dx u_3 = - \int_{x_1}^{x_3} \varphi_2 dx \end{aligned} \quad \text{for } k=2 \text{ and} \quad (6.35)$$

$$\begin{aligned} & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_2 dx u_2 + \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_3 dx u_3 - \int_{x_2}^{x_3=1} \varphi_3 \varphi_2 dx u_2 - \int_{x_2}^{x_3=1} \varphi_3 \varphi_3 dx u_3 = \\ & [\varphi_3 \cdot 1]_{x=1} - \int_{x_2}^{x_3=1} \varphi_3 dx \end{aligned} \quad \text{for } k=3. \quad (6.36)$$

We will set up the relation above in matrix format as

$$\mathbf{KU} - \mathbf{MU} = \mathbf{F}_{ext}. \quad (6.37)$$

The expression becomes

$$\begin{aligned} & \left[\begin{array}{ccc} \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_1 dx & \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_2 dx & 0 \\ \int_{x_1}^{x_3} \varphi'_2 \varphi'_1 dx & \int_{x_1}^{x_3} \varphi'_2 \varphi'_2 dx & \int_{x_1}^{x_3} \varphi'_2 \varphi'_3 dx \\ 0 & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_2 dx & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_3 dx \end{array} \right] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \\ & \left[\begin{array}{ccc} \int_{x_1=0}^{x_2} \varphi_1 \varphi_1 dx & \int_{x_1=0}^{x_2} \varphi_1 \varphi_2 dx & 0 \\ \int_{x_1}^{x_3} \varphi_2 \varphi_1 dx & \int_{x_1}^{x_3} \varphi_2 \varphi_2 dx & \int_{x_1}^{x_3} \varphi_2 \varphi_3 dx \\ 0 & \int_{x_2}^{x_3=1} \varphi_3 \varphi_2 dx & \int_{x_2}^{x_3=1} \varphi_3 \varphi_3 dx \end{array} \right] = - \left[\begin{array}{c} \int_{x_1=0}^{x_2} \varphi_1 dx \\ \int_{x_1}^{x_3} \varphi_2 dx \\ \int_{x_2}^{x_3=1} \varphi_3 dx \end{array} \right] + \left[\begin{array}{c} 0 \\ 0 \\ [\varphi_3 \cdot 1]_{x=1} \end{array} \right] \end{aligned} \quad (6.38)$$

We split the integrals in \mathbf{K} over intervals between nodes in order to make it clear why element based formulations are more effective. We get

$$\mathbf{K} =$$

$$\begin{bmatrix} \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_1 dx & \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_2 dx & 0 \\ \left[\int_{x_1=0}^{x_2} \varphi'_2 \varphi'_1 dx + \int_{x_2}^{x_3=1} \varphi'_2 \varphi'_1 dx \right] & \left[\int_{x_1=0}^{x_2} \varphi'_2 \varphi'_2 dx + \int_{x_2}^{x_3=1} \varphi'_2 \varphi'_2 dx \right] & \left[\int_{x_1=0}^{x_2} \varphi'_2 \varphi'_3 dx + \int_{x_2}^{x_3=1} \varphi'_2 \varphi'_3 dx \right] \\ 0 & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_2 dx & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_3 dx \end{bmatrix}. \quad (6.39)$$

Some of the integrals are zero as the functions do not overlap. This gives

$$\mathbf{K} =$$

$$\begin{bmatrix} \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_1 dx & \int_{x_1=0}^{x_2} \varphi'_1 \varphi'_2 dx & 0 \\ \int_{x_1=0}^{x_2} \varphi'_2 \varphi'_1 dx & \left[\int_{x_1=0}^{x_2} \varphi'_2 \varphi'_2 dx + \int_{x_2}^{x_3=1} \varphi'_2 \varphi'_2 dx \right] & \int_{x_2}^{x_3=1} \varphi'_2 \varphi'_3 dx \\ 0 & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_2 dx & \int_{x_2}^{x_3=1} \varphi'_3 \varphi'_3 dx \end{bmatrix}. \quad (6.40)$$

There is a drawback with evaluating the integrals in the matrices according to Eq. (6.38). This is seen for the case above. For example, evaluating K_{11} requires setting up information about the integral from x_1 to x_2 . Then this information has to be set up repeatedly for K_{12} and K_{22} . It is more efficient to first evaluate all contributions for the interval integral from x_1 to x_2 and thereafter integral from x_2 to x_3 . Then each subintegral is summed into the appropriate position in \mathbf{K} as there may be values there already from previous domains.

The split of the integrals into element integrals requires that there is some kind of minimum continuity required for the integrands in order to be permissible. The integrand must not be infinite at the boundary between the elements. This places a requirement both on the interpolation of the primary variables over the elements as well as how elements are combined for a problem.

We will introduce new notations that will be the basis for element based finite element formulation according to next section. The finite element method is implemented with a loop over elements where element matrices and vectors are evaluated and then summed (assembled) into the global matrices and vectors.

We split the nodal based functions for the centre node into two parts as in the figure below. They will be called *shape functions* and are denoted with N :s. The right subscript is 1 or 2 denoting a slope downhill respectively uphill in positive x-direction. The right superscript denotes the number of the *element*. It is element number 1 from x_1 to x_2 and 2 from x_2 to x_3 . We will later skip this superscript.

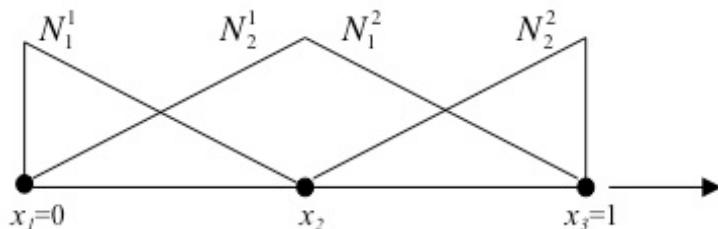


Figure 6.5. Three nodes used to solve Eq.s (6.1)-(6.3).

The contributions to \mathbf{K} in Eq. (6.40) can now be written as

$$\mathbf{K} = \begin{bmatrix} \int_{x_1=0}^{x_2} \frac{dN_1^1}{dx} \frac{dN_1^1}{dx} dx & \int_{x_1=0}^{x_2} \frac{dN_1^1}{dx} \frac{dN_2^1}{dx} dx & 0 \\ \int_{x_1=0}^{x_2} \frac{dN_2^1}{dx} \frac{dN_1^1}{dx} dx & \left[\int_{x_1=0}^{x_2} \frac{dN_2^1}{dx} \frac{dN_2^1}{dx} dx + \int_{x_2}^{x_3=1} \frac{dN_1^2}{dx} \frac{dN_1^2}{dx} dx \right] & \int_{x_2}^{x_3=1} \frac{dN_1^2}{dx} \frac{dN_2^2}{dx} dx \\ 0 & \int_{x_2}^{x_3=1} \frac{dN_2^2}{dx} \frac{dN_1^2}{dx} dx & \int_{x_2}^{x_3=1} \frac{dN_2^2}{dx} \frac{dN_2^2}{dx} dx \end{bmatrix}, \quad (6.41)$$

or even shorter

$$\mathbf{K} = \begin{bmatrix} k_{11}^1 & k_{12}^1 & 0 \\ k_{12}^1 & k_{22}^1 + k_{11}^2 & k_{12}^2 \\ 0 & k_{21}^2 & k_{22}^2 \end{bmatrix}, \quad (6.42)$$

The matrices \mathbf{K} and \mathbf{M} are all symmetric. For example we have $k_{12}^1 = k_{21}^1$. It is also clear why the diagonal elements always are positive. We will show how to perform the element integrals in section 6.4 but first some note about the assembly procedure.

6.3 Assembly procedure

The previous example indicated that the matrix \mathbf{K} was banded. The shape functions that did not overlap gave no contributions. It becomes filled along the diagonal and the nearest off-diagonals. K_{13} was zero in the previous example. The general property of finite element matrices is that they are not full due to the use of local functions. However, they need not have the nice tridiagonal structure as in the example above.

Different ordering of the node numbers, or rather equation numbers²⁰, will change where different values will be placed in the matrix. The diagonal number couples the response at a given node with the input to that node. The use of local functions leads to that only nodes that have functions that overlap will contribute to the integrals and thereby to the numbers at specific positions in the matrix \mathbf{K} . The two examples in Figure 6.6 fill the matrix \mathbf{K} differently. The top example in the figure gives

$$\mathbf{K} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \quad (6.43)$$

whereas the lower case gives

$$\mathbf{K} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & 0 & x \\ 0 & 0 & x & x \\ 0 & x & x & x \end{bmatrix} \quad (6.44)$$

²⁰ Each node has only one unknown value and therefore nodal numbers and equation numbers can be the same.

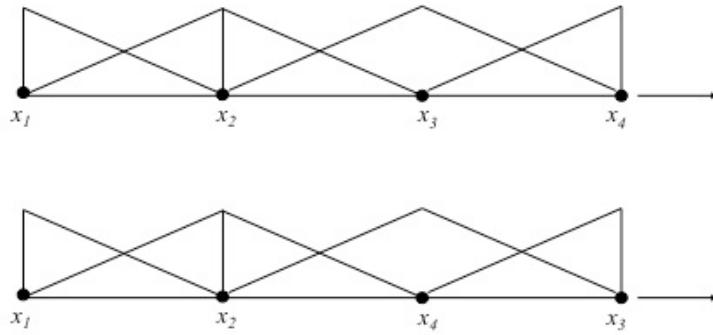


Figure 6.6. Two examples of numbering of nodes.

The number at each position (k,j) in the matrix can be interpreted in terms of the physics of the problem. One obvious interpretation is that this number give the contribution from the j :th degree of freedom, u_j , to the k :th nodal balance law equation. In mechanics it is understood as the contribution from the displacement u_j to the nodal equilibrium for the k :th degree of freedom. Then the field $u(x)$ in Eq. (6.1) is a displacement field and the equation we solve is an equilibrium equation expressed in the displacements. The left end has a given value (displacement in mechanics) fulfilled in the solution by setting $u_1=0$. The right end has a given flux given by Eq. (6.3) to 1. This value enters into the equilibrium equation as a 1 on the right hand side, last term, in Eq. (6.38) for $k=N=3$. The formulation approximates the balance law (flux) at the right end of the domain and it enters as a ‘load’ in the nodal balance equations.

More about the assembly procedure and how it is automated for coding is shown at end of section 6.4.

6.4 Element based trial and weight functions

The observations in the previous section is the motivation for making the element the primary entity and describe the total field by splitting the domain into elements and interpolating the primary variable over elements and the elements are used to discretise the whole domain. Then we get

$$\hat{u}(x) = \sum_{e=1}^{N_{elem}} u^e(x), \quad (6.45)$$

where N_{elem} is the number of elements used to solve the problem. The field within an element is written as

$$u^e(x) = \sum_{i=1}^{N_{node}} N_i u_i = \mathbf{N} \mathbf{u}, \quad (6.46)$$

where N_{node} is the number of nodes associated with the element. We do not include and right superscript with element number on \mathbf{N} . Now we change the notation once more for the trial functions as we switch to element description. Eq. (6.28) can be written for our earlier example as

$$\mathbf{N}(x) \mathbf{u} = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} x_2 - x \\ l^e & x - x_1 \\ l^e \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (6.47)$$

where $l^e = x_2 - x_1$ is the element length. This is the same as our dx used earlier. \mathbf{N} is the shape function matrix and N_i is the shape function associated with local node number i of the element. The shape functions are the most important property of an element. The element is shown in Figure 6.7. Notice that we have introduced local node numbers, 1 and 2, denoting in

this case the left and right node, respectively. Now it will be very important to keep two systems of notations apart, the large structure with its numbers and relations etc versus the local element variables and relations. We will use small letters²¹ for local element matrices and vectors and capital for those associated with the analysed problem. The two node element is a linear element as it has only two degrees of freedoms for the element (u_1 and u_2) and thereby one can only describe a linear variation of the field inside the element. Therefore, it can be called a linear element in this respect (although it can be used to solve nonlinear problems).

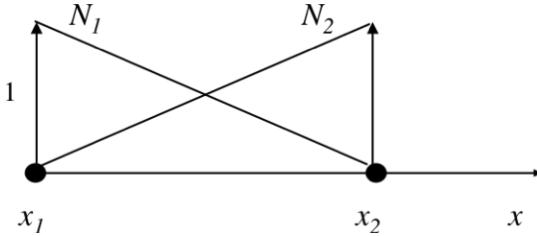


Figure 6.7. Two node element and its linear shape functions N_1 and N_2 .

We will rewrite the element by introducing a local coordinate system also. This is not necessary for this simple one-dimensional element but extremely important in the general case. Then the element integrals can be integrated over a simple region, which enables the use of elements in two and three dimensions without having to stay with rectangular shapes. At first sight this complicates the element formulation as this variable change makes it impossible to solve the integrals analytically in the general case. However, the need for numerical integration to evaluate these integrals is not only very effective but also gives an additional possibility to develop elements with different properties. Thus this seemingly drawback is also a possibility in element formulation.

We start with the element formulation in the local coordinate system and will in the end describe the overall logic corresponding to the nodal based approach in section 6.2.

The element is described in a local coordinate system, s , as shown in Figure 6.8.

$$u(s(x)) = \mathbf{N}(s)\mathbf{u} = \begin{bmatrix} \frac{1-s}{2} & \frac{1+s}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (6.48)$$

This variable change can be described in the same way

$$x(s) = \mathbf{N}(s)\mathbf{x} = \begin{bmatrix} \frac{1-s}{2} & \frac{1+s}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (6.49)$$

This is an example of an isoparametric element as the same parameters used for interpolation the unknown field is used to interpolate the geometry between the nodes. There exists also super- and sub-parametric elements but isoparametric is the most common.

²¹ Although \mathbf{N} and later introduced matrix \mathbf{B} are block letters they always refer to elements. This is a quite common notation in FE-literature.

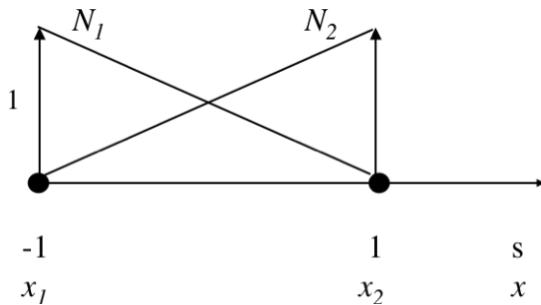


Figure 6.8. Two node element and its linear shape functions N_1 and N_2 with a local coordinate system, s .

The integrals in Eq. (6.25) will now be integrated elementwise²² as

$$\int_{x_1}^{x_2} \left(\frac{d\mathbf{N}}{dx} \right)^T \frac{d\mathbf{N}}{dx} dx \mathbf{u} - \int_{x_1}^{x_2} (\mathbf{N}^T \mathbf{N} \mathbf{u} - \mathbf{N}^T \cdot \mathbf{1}) dx. \quad (6.50)$$

Thus Eq. (6.50) includes the contribution from the trial and weight functions of these two nodes to the overall system of equations needed. The first integral in Eq. (6.50) can be written as

$$\mathbf{k}^{(1)} = \int_{x_1}^{x_2} \begin{bmatrix} \frac{dN_1}{dx} & \frac{dN_2}{dx} \\ \frac{dN_2}{dx} & \frac{dN_1}{dx} \end{bmatrix} \begin{bmatrix} \frac{dN_1}{dx} & \frac{dN_1}{dx} \\ \frac{dN_2}{dx} & \frac{dN_2}{dx} \end{bmatrix} dx = \int_{x_1}^{x_2} \begin{bmatrix} \frac{dN_1}{dx} \frac{dN_1}{dx} & \frac{dN_1}{dx} \frac{dN_2}{dx} \\ \frac{dN_2}{dx} \frac{dN_1}{dx} & \frac{dN_2}{dx} \frac{dN_2}{dx} \end{bmatrix} dx = \begin{bmatrix} k_{11}^1 & k_{12}^1 \\ k_{21}^1 & k_{22}^1 \end{bmatrix}. \quad (6.51)$$

This is the same as the upper left 2×2 part in Eq. (6.42). We introduce a specific notation for the $\frac{d\mathbf{N}}{dx}$ matrix as it is commonly used relating primary variables to gradients. We denote it by

$$\mathbf{B} = \frac{d\mathbf{N}}{dx}. \quad (6.52)$$

Thus we write

$$\int_{x_1}^{x_2} \mathbf{B}^T \mathbf{B} dx \mathbf{u} - \int_{x_1}^{x_2} \mathbf{N}^T \mathbf{N} dx \mathbf{u} + \int_{x_1}^{x_2} \mathbf{N}^T \cdot \mathbf{1} dx. \quad (6.53)$$

Notice that we contribute to two balance equations, two k :s, at the same time as \mathbf{B}^T and \mathbf{N}^T has the weight functions for both nodes of the element. What k :s depend on which global node number the local node number 1 has and same for local node number 2. Furthermore, the integral only contributes to two j :s corresponding to u_1 and u_2 and therefore we do not set the equation above equal to zero. The j :s thus are the same as the k :s!

We need to extract some information about the variable change before embarking on the equation above. The equation requires the derivative w.r.t x and not s . We must change the variable limits and the integration parameter dx . All this has to do with the change of scale when changing coordinate system. The chain rule gives

$$\frac{d\mathbf{N}(s(x))}{dx} = \frac{d\mathbf{N}}{ds} \frac{ds}{dx}. \quad (6.54)$$

²² We do not write equal zero as this formula is only for one element and is only a subset of a balance equation.

Eq. (6.49) gives

$$\frac{dx}{ds} = J = \frac{d\mathbf{N}}{ds} \mathbf{x} = \begin{bmatrix} -1 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{x_2 - x_1}{2} = \frac{l^e}{2}, \quad (6.55)$$

where J is the Jacobian of the mapping (or variable change) that is the change in scale. Inverting this gives

$$\frac{ds}{dx} = J^{-1} = \frac{2}{l^e}. \quad (6.56)$$

Now we can write

$$\mathbf{B} = \frac{d\mathbf{N}}{dx} = \frac{d\mathbf{N}}{ds} J^{-1} = \begin{bmatrix} -1 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \frac{2}{l^e} = \begin{bmatrix} -1 & 1 \\ \frac{1}{l^e} & \frac{1}{l^e} \end{bmatrix}. \quad (6.57)$$

This answer is what anticipated. The shape functions change from 0 to 1 over the length l^e . Now we are ready to rewrite our element integrals Eq. (6.53) in the local coordinate system. We get

$$\int_{-1}^1 \mathbf{B}^T \mathbf{B} J ds \mathbf{u} - \int_{-1}^1 \mathbf{N}^T \mathbf{N} J ds \mathbf{u} + \int_{-1}^1 \mathbf{N}^T \cdot \mathbf{1} J ds. \quad (6.58)$$

The relation above can be expressed in local, element matrices and vectors as

$$\mathbf{k}_1 \mathbf{u} - \mathbf{k}_2 \mathbf{u} - \mathbf{f}_{ext} = \mathbf{k} \mathbf{u} - \mathbf{f}_{ext}, \quad (6.59)$$

where

$$\mathbf{k}_1 = \int_{-1}^1 \begin{bmatrix} -1 \\ \frac{1}{l^e} \\ \frac{1}{l^e} \end{bmatrix} \left[\begin{bmatrix} -1 & 1 \\ \frac{1}{l^e} & \frac{1}{l^e} \end{bmatrix} \frac{l^e}{2} \right] ds = \int_{-1}^1 \begin{bmatrix} \frac{1}{l^2} & -\frac{1}{l^2} \\ -\frac{1}{l^2} & \frac{1}{l^2} \end{bmatrix} \frac{l^e}{2} ds = \begin{bmatrix} \frac{1}{l^e} & -\frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{l^e} \end{bmatrix} \quad (6.60)$$

and

$$\mathbf{k}_2 = \int_{-1}^1 \begin{bmatrix} \frac{1-s}{2} \\ \frac{1+s}{2} \\ \frac{1+s}{2} \end{bmatrix} \left[\begin{bmatrix} 1-s & 1+s \\ 2 & 2 \\ 2 & 2 \end{bmatrix} \frac{l^e}{2} \right] ds = \int_{-1}^1 \begin{bmatrix} \frac{(1-s)^2}{4} & \frac{1-s^2}{4} \\ \frac{1-s^2}{4} & \frac{(1+s)^2}{4} \end{bmatrix} \frac{l^e}{2} ds = \begin{bmatrix} \frac{l^e}{3} & \frac{l^e}{6} \\ \frac{l^e}{6} & \frac{l^e}{3} \end{bmatrix} \quad (6.61)$$

This gives

$$\mathbf{k} = \begin{bmatrix} \frac{1}{l^e} & -\frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{l^e} \end{bmatrix} - \begin{bmatrix} \frac{l^e}{3} & \frac{l^e}{6} \\ \frac{l^e}{6} & \frac{l^e}{3} \end{bmatrix} \text{ and} \quad (6.62)$$

$$\mathbf{f}_{ext} = - \int_{-1}^1 \mathbf{N}^T \cdot \mathbf{1} J ds = - \int_{-1}^1 \begin{bmatrix} \frac{1-s}{2} \\ \frac{1+s}{2} \\ \frac{1+s}{2} \end{bmatrix} \frac{l^e}{2} ds = - \begin{bmatrix} \frac{l^e}{2} \\ \frac{l^e}{2} \end{bmatrix}. \quad (6.63)$$

These are the same results as in the previous section expressed in another format. We use small letters to indicate that these are element matrices. They will give the same resulting system of equations after what is called the assembly procedure. The latter is a step that was

not needed in the nodal based approach as we did directly form the global matrices and no element variables where we have a local numbering system, in this case from 1 to 2. As stated earlier - we need to identify for each element which global (j,k) our nodes correspond to. We will illustrate this in Table 6.3 below by making an assembly of the using the upper model in Figure 6.6. This is demonstrated for the lower model in Figure 6.6 in Table 6.4 but then in a way that is another step towards what is done in an actual FE-software.

Table 6.3. Assembly procedure.

Logic for element assembly

Zero global matrix \mathbf{K} , a 4×4 matrix, and global vector \mathbf{F}_{ext} , a 4×1 vector.

Element 1 has node number 1 as local node number 1 and node number 2 as right node. Thus we add the matrix \mathbf{k} according to Eq (6.62) to \mathbf{K} and vector \mathbf{f}_{ext} to \mathbf{F}_{ext} . Then we get

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Element 2 has node number 2 as left node 1 and node number 3 as local node number 2. Thus we add the matrix \mathbf{k} according to Eq. (6.62) to \mathbf{K} and vector \mathbf{f}_{ext} to \mathbf{F}_{ext} . Then we get

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 \\ 0 & -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Element 3 has node number 3 as left node 1 and node number 4 as right node. Thus we add the matrix \mathbf{k} according to Eq. (6.62) to \mathbf{K} and vector \mathbf{f}_{ext} to \mathbf{F}_{ext} . Then we get

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 \\ 0 & -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} \\ 0 & 0 & -\frac{1}{l^e} - \frac{l^e}{6} & -\frac{1}{l^e} - \frac{l^e}{6} \end{bmatrix} \quad \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

This is the same as would be obtained using the formula in Eq. **Error! Reference source not found.**

Table 6.4. Automated assembly procedure.

Logic for element assembly

Zero global matrix \mathbf{K} , a 4x4 matrix, and global vector \mathbf{F}_{ext} , a 4x1 vector.

Element 1

Set up identification vector where the i :th position gives the global equation number of the degree of freedom that has local number i . It is

$$\mathbf{id} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

This is used to automatically locate where corresponding rows/columns of local matrices/vectors are in the global ones. Thus we use place this as transpose above the matrix to identify global column numbers and to the right to identify global row numbers where corresponding values should be assembled. Thus element one with global node numbers 1 and 2 has

$$\mathbf{k} = \begin{bmatrix} 1 & 2 \\ \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \mathbf{f}_{ext} = -\begin{bmatrix} \frac{l^e}{2} \\ \frac{l^e}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Then we get

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Element 2

$$\mathbf{id} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

with

$$\mathbf{k} = \begin{bmatrix} 2 & 4 \\ \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \mathbf{f}_{ext} = -\begin{bmatrix} \frac{l^e}{2} \\ \frac{l^e}{2} \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Giving

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} & 0 & -\frac{1}{l^e} - \frac{l^e}{6} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & \frac{1}{l^e} - \frac{l^e}{3} \end{bmatrix} \quad \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Element 3

$$\mathbf{id} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

with

$$\mathbf{k} = \begin{bmatrix} 4 & 3 \end{bmatrix} \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{1}{l^e} - \frac{l^e}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \mathbf{f}_{ext} = -\begin{bmatrix} \frac{l^e}{2} \\ \frac{l^e}{2} \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

Giving

$$\mathbf{K} = \begin{bmatrix} \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} & 0 & 0 \\ -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} & 0 & -\frac{1}{l^e} - \frac{l^e}{6} \\ 0 & 0 & \frac{1}{l^e} - \frac{l^e}{3} & -\frac{1}{l^e} - \frac{l^e}{6} \\ 0 & -\frac{1}{l^e} - \frac{l^e}{6} & -\frac{1}{l^e} - \frac{l^e}{6} & \frac{2}{l^e} - \frac{2l^e}{3} \end{bmatrix} \quad \mathbf{F}_{ext} = -\frac{l^e}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

Imposing the essential boundary condition is done principally in the same way as for the earlier cases by prescribing $u_1=1$. The results shown are shown in Table 6.5. One can note that it converges slower than when using global trial functions. See also the discussion in chapter 5. However, the condition number increases much slower using the FE-approach. The condition number is 3.4e8 and the error 5.1e-9 is when using $N=10000$.

Table 6.5. L2-error and condition number of matrix for Finite Element Method.

Number of nodes (N)	Error	Condition number	Number of terms (N)	Error	Condition number
2	0.015	1.0000	6	0.00078	99.8
3	0.0046	13.9	7	0.00055	140.
4	0.0021	39.1	8	0.00040	188.
5	0.0012	66.1	9	0.00031	242

Table 6.6. Matlab™ code for the Finite Element Method.

```

Fext(1)=-dx*0.5;           % Eq. Error! Reference source not found.a
Fext(N)=1-dx*0.5;          % Eq. Error! Reference source not found.b
invdx=1/dx;
K(1,1)=invdx-dx/3         % Eq. Error! Reference source not found.
K(N,N)=K(1,1);             % Eq. Error! Reference source not found.
kdiag=2*K(1,1);
koffdiag=-1/dx-dx/6;
K(1,2)=koffdiag;
K(2,1)=koffdiag;
for k=2:N-1
    Fext(k)=-dx;           % Eq. Error! Reference source not found.
    K(k,k)=kdiag;           % Eq. Error! Reference source not found.
    K(k,k+1)=koffdiag;      % Eq. Error! Reference source not found.
    K(k+1,k)=koffdiag;      % Eq. Error! Reference source not found.
end
% We impose the condition a1=1 that is multiplying first column of K
a1=1;
Fmod=K(:,1)*a1;
% Move this to right hand side
F=Fext-Fmod;
% The first equation for a1 is not needed any more
F(1)=[];K(1,:)=[];K(:,1)=[];
a2=1/cos(1);
Fmod=K(:,1)*a2;
% Move this to right hand side
F=F-Fmod;
% The current first equation for a2 is not needed any more
F(1)=[];K(1,:)=[];K(:,1)=[];

```

6.5 Processing of finite element results

The processing of the results from a finite element solution is often called post-processing. Creating of the mesh before the finite element analysis is called pre-processing. Visualisation of computed nodal values is quite straightforward. However, plotting of gradients and flux requires some attention. Flux is related to the gradient by a constitutive model.

The finite element shape functions ensures continuous primary variables between elements but usually not for the gradients. We will illustrate this with the following problem. It is a simpler version of the problem in section 4.8. The equations are

$$L(u) = EA \frac{d^2u}{dx^2} + A\rho g = 0 \quad \text{where } x \in [0, L] \quad \text{with boundary conditions} \quad (6.64)$$

$$B_e(0) = u(0) = 0 \quad \text{and} \quad (6.65)$$

$$B_n(L) = \left. \frac{du}{dx} \right|_{x=L} = 0. \quad (6.66)$$

The following physical interpretation can be done for the problem Eq. (6.64) is the equilibrium equation for a hanging rod expressed in its axial displacement u . It has the cross-sectional area A and E is its elasticity modulus. ρ is the density and g is the gravity constant. The rod is fixed at $x=0$ and free at the other end.

The exact solution is

$$u(x) = \frac{\rho g L^2}{2E} \frac{x}{L} \left[2 - \frac{x}{L} \right] \quad (6.67)$$

The gradient of the displacement is called strain ε and the flux is the stress σ where $\sigma = E\varepsilon$. They stress is

$$\sigma = E\varepsilon = E \frac{du}{dx} = \rho g L \left[1 - \frac{x}{L} \right] \quad (6.68)$$

We will use the two node element derived previously to solve this problem. A comparison between Eq. (6.64) and Eq. (6.1) leads to that Eq. (6.58) now becomes

$$\int_{-1}^1 EAB^T \mathbf{B}J ds \mathbf{u} - \int_{-1}^1 \mathbf{N}^T \cdot \rho Ag J ds. \quad (6.69)$$

The matrix in Eq. (6.60), called stiffness matrix for mechanical problems, becomes

$$\mathbf{k} = \frac{EA}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \quad (6.70)$$

The consistent nodal load, Eq. (6.63)

$$\mathbf{f}_{ext} = \frac{\rho Ag l^e}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{m^e g}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (6.71)$$

A solution using two elements is shown in Figure 6.9. The finite elements has one degree too low polynomial in the shape function in order to match the exact solution. However, it is correct at the nodes. The stresses are shown in Figure 6.10. Two variants of the computed stresses are shown. The stair like curve is the stresses in the two elements. They are constant at each element. The exact curve is near these values at the centre of each element. The nodal average curve looks more like the exact solution. The stress at each node is taken as the average of the surrounding element's stresses. A linear variation is plotted between the nodes. The element stresses are best at element interiors, more precisely at the integration points introduce in chapter 7. The element stresses tend to oscillate around the exact solution and therefore a smoothed solution gives a better view of the results. It can also be noted that the discontinuity in stresses between the elements are due to the discretisation error. The jumps will decrease when more elements are used. Therefore, this information is used in adaptive finite element codes to estimate where the mesh need be refined. Table 6.7 shows various error measures and condition number of stiffness matrix for increasing number of elements. The L2-error for nodal displacement is evaluated over the whole domain using 50 locations. It would be exact zero if only evaluated at the nodes. Notice the larger error for stresses.

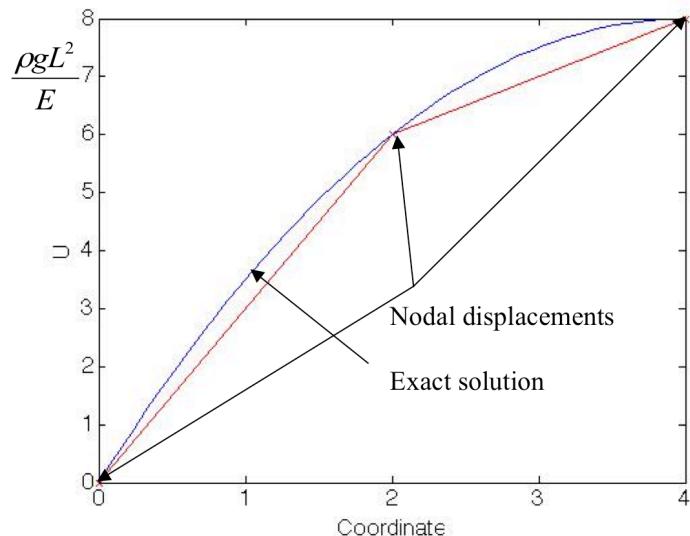


Figure 6.9. Comparison between finite element and exact displacements for hanging rod using two elements.

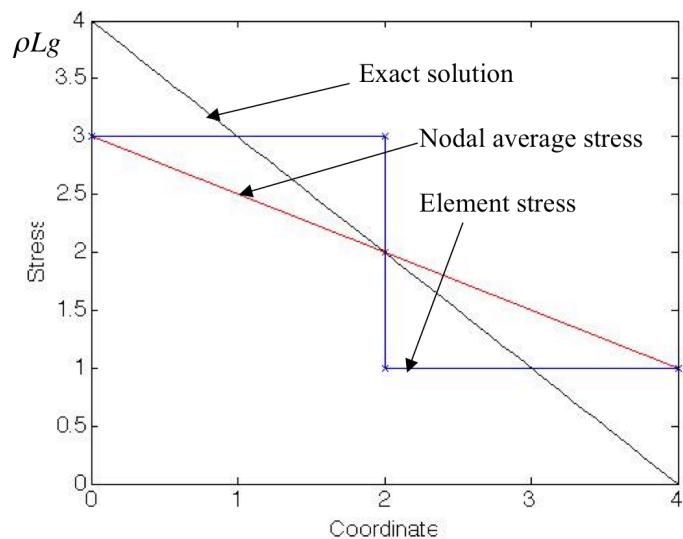


Figure 6.10. Comparison between finite element and exact stresses for hanging rod.

Table 6.7. L2-error of nodal displacements, element stresses and smoothed nodal stresses.

Number of elements	Displacement error	Element stress error	Smoothed nodal stress error	Condition number
2	0.052	0.667	0.471	9
4	0.013	0.202	0.141	40
8	0.0032	0.066	0.039	144
16	0.0008	0.023	0.010	544
32	0.0002	0.008	0.003	2112

6.6 Exercises

Exercise 6.1

The Tonti diagram in Figure 3.2 can be applied to varying physical problems like electrostatics²³ in the current problem. The task is to set up an element electric conductivity matrix for a steady state current. We limit the case to flow along wires, i.e. uniaxial relations. We replace u , g , q and f by the more common symbols in electrostatics and the Tonti diagram becomes as below.

The electric field E is a conservative field, i.e. no work is done for a closed loop of motion of a charge. This is the basis for being able to introduce a potential function Φ where the electric field is its gradient as $E = -\frac{d\Phi}{dx}$.

The electric field is related to the current flowing in the material. The electric displacement, D , is obtained by a constitutive relation from the electric field by $D = \epsilon E$.

The current density²⁴ is $J = J_f + J_b + J_D = J_f + J_b + \frac{dD}{dt} \approx \sigma E$, where σ is the electric conductivity of the material. The electric displacement (motion of charges) is by a balance law related to the charge density ρ as $\rho = \frac{dD}{dx}$.

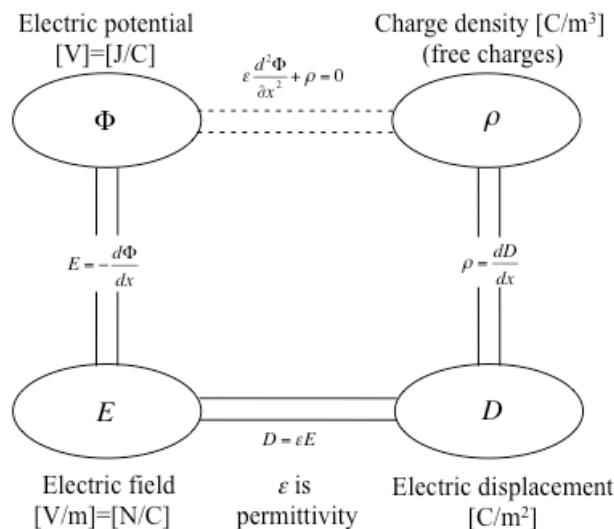


Figure 6.11. Tonti diagram for electrostatics.

- Set up the diagram for a two node rod, corresponding to Figure 3.6, that will be used to solve electrostatic problems. The rod has length l^e .
- Use the diagram to obtain the element conductivity matrix directly.
- Use the element based approach in section 6.4 to derive the same conductivity matrix by use of the differential equation $\epsilon \frac{d^2\Phi}{dx^2} + \rho = 0$ instead of Eq. (6.1). Naturally the answer is the same as for case b.

Answer: for b) and c) is $\mathbf{k} = \int_{l^e} \mathbf{B}^T \mathbf{B} \epsilon dx = \frac{\epsilon}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

²³ <http://en.wikipedia.org/wiki/Electrostatics>

²⁴ http://en.wikipedia.org/wiki/Current_density

Exercise 6.2

The previous formulation can be rewritten to relations between electric potential Φ and current running in a wire, I . You may have use for Ohms law $RI = \Delta\Phi$ leading to

$$I = AJ = A\rho v_s = \frac{\Delta\Phi}{R} = \frac{A\sigma}{l^e} \Delta\Phi.$$

where J is the flux per unit volume and I is the current through a wire with cross-section A . σ is the electric conductivity of the material. R is the resistance of the wire and $\Delta\Phi$ is its voltage drop. The latter is $\Phi_2 - \Phi_1$ in case of a rod. σ is the electric conductivity of the material in the wire.

The resulting finite element system of equations express continuity of current, i.e. Ampere's circuital law.

Answer: $\frac{1}{R} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$.

Exercise 6.3 (Computational or manual solution)

Calculate the electric potentials at the nodes of the network below using the derived element matrix from Exercise 6.1. Compute the nodal current at first and last nodes (no 1 and 5).

The technique in Table 6.4 can be used for assembling element contributions to global system to be solved. Notice that one potential is prescribed to zero (ground) and the other is a non-zero value.

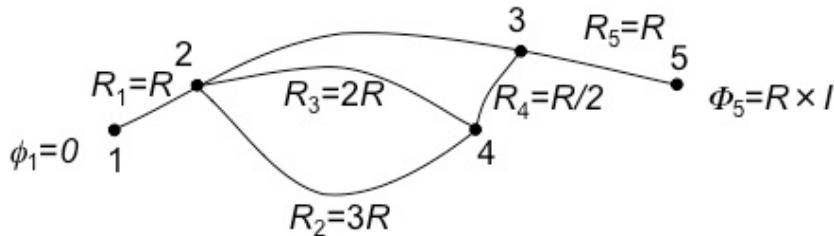


Figure 6.12. Network for flow of current. Figure indicates possible use of node and element numbers. The first node is grounded (0V) and the last has also a prescribed potential. R denotes a resistance and I is current.

Answer: $I_5 = -I_1 = \frac{10}{37}I$. Current flows in at node 5 and out of node 1. Potential increases

from 1 to 5. The intermediate node (we have prescribed $\phi_1=0$ and $\phi_5=R\times I$) potentials are

$$\begin{bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \frac{RI}{37} \begin{bmatrix} 10 \\ 27 \\ 22 \end{bmatrix}, \text{ i.e. increasing towards node 5.}$$

Exercise 6.4

The equations for potential flow²⁵ in tubes is similar to the case in Exercise 6.2 written as

²⁵ http://en.wikipedia.org/wiki/Potential_flow

$$\frac{d^2\Phi}{dx^2} = 0$$

The velocity of the flow is the gradient of the potential field $v = \frac{d\Phi}{dx}$

Write the element matrix relating potential drop if the mass flow through a pipe is related to the pressure (p) drop by $R_{\text{pipe}}v = \Delta p = p_2 - p_1$. This relation is an outcome of Darcys law²⁶.

Answer: $\frac{1}{R(\text{Re})_{\text{pipe}}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$. We have indicated that the permability of the

pipe, R_{pipe} , does not only depend on the pipe geometry and surface but also on the fluid itself, its viscosity and velocity. The Reynolds number (Re) determines what kind of fluid flow there will be (laminar, turbulent).

Exercise 6.5 (Computational problem)

Consider in $x \in [0, L]$, the second order Boundary Value Problem

$$\lambda \frac{d^2u}{dx^2} + q = 0$$

where $q=a+bx$. The solution is subject to the boundary conditions

$$u(0) = 0 \text{ and } \left. \frac{du}{dx} \right|_{x=L} = 0$$

Find an approximate solution using the using a three-node element.

The shape function of the element is, in a local coordinate system $s \in [-1, 1]$,

$$\mathbf{N} = \begin{bmatrix} -\frac{1}{2}s(1-s) & (1-s^2) & \frac{1}{2}s(1+s) \end{bmatrix}$$

Thus local node number 1 is to the left ($s=-1$) and number 2 is in centre ($s=0$) and the third node is to the right ($s=1$).

Hint: Assume that the centre node is always in the middle

$$x_2 = \frac{x_1 + x_3}{2}$$

This gives a constant Jacobian, same as two-node element in Eq. (6.56).

Implement the solution in Matlab™ and evaluate it with respect to the exact solution. Make the comparison both for u itself as well as its first derivative, $\frac{du}{dx}$. Perform a convergence study like, for example, in Table 6.5.

Exercise 6.6 (Computational problem)

Consider in $x \in [0, L]$, the second order Boundary Value Problem

$$EA \frac{d^2u}{dx^2} + P\delta(x-x_i) = 0$$

, where $\delta(x-x_i)$ Tis the Dirac function at $x=x_i$. The solution is subject to the boundary conditions

²⁶ http://en.wikipedia.org/wiki/Darcy's_law

$$u(0) = 0 \text{ and } \left. \frac{du}{dx} \right|_{x=L} = \frac{F}{EA}$$

Find an approximate solution using the using a two node element as described in 6.4.
Implement the solution in Matlab™ and evaluate it with respect to the exact solution. Make
the comparison both for u itself as well as its first derivative, $\frac{du}{dx}$. Perform a convergence
study like, for example, in Table 6.5. The point x_i with the point load can be chosen freely but
two finite element solutions should be made. One with x_i inside an element and one where x_i
coincides with a location of a node.

7 Numerical integration

The previous chapter brought us to isoparametric element but only in the one-dimensional context. Next chapters generalise this to multidimensional problems where it will be clear that we cannot form an analytic solution of the element integrals. Thus we prepare by introducing the concept of numerical integration^{27,28}.

A one-dimensional integral over the domain [-1,1] of a polynomial, f , can be integrated exactly by the formula

$$I = \int_{-1}^1 f ds = \sum_{i=1}^{nint} f(s_i) w_i \quad (7.1)$$

where s_i are the integration points, w_i are their weights and $nint$ are the number of integration points. The most common rule is Gauss' rule. It is the most effective for one-dimensional integrals. There are other rules, for example the Lobatto rules that have points at end of interval. However, we discuss only the Gauss integration rule here. It is the most common and therefore one often uses the notation Gausspoints about the sampling point for the integrand f . The higher the polynomial to integrate, the more integration points are needed. The Gauss integration rule is shown in Table 7.1. The relation between the degree of the polynomial, n , that can be exact integrated and the number of integration points, $nint$, is

$$n = 2 \cdot nint - 1 \quad (7.2)$$

Table 7.1. Gauss integration rule.

$Nint$	Integration points	Weights
1	0	2
2	$-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$	1,1
3	$-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}$	$\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$

The rule in Table 7.1 is used to integrate a cubic polynomial. The analytic solution is

$$I = \int_{-1}^1 (a + bs + cs^2 + ds^3) ds = \left[as + \frac{bs^2}{2} + \frac{cs^3}{3} + \frac{ds^4}{4} \right]_{-1}^1 = 2a + \frac{2c}{3} \quad (7.3)$$

The needed two point rule gives

$$I = f\left(\frac{1}{\sqrt{3}}\right) \cdot 1 + f\left(-\frac{1}{\sqrt{3}}\right) \cdot 1 = 2a + \frac{2c}{3} \quad (7.4)$$

This rule is applied ‘cross-wise’ for two- and tree-dimensional problems by

$$I = \int_{-1}^1 \int_{-1}^1 f ds dt = \sum_{i=1}^{nint} \sum_{j=1}^{nint} f(s_i, t_j) w_i w_j \quad (7.5)$$

²⁷ http://en.wikipedia.org/wiki/Numerical_integration

²⁸ http://en.wikipedia.org/wiki/Gaussian_quadrature

or

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f \, ds \, dt \, dz = \sum_{i=1}^{nint} \sum_{j=1}^{nint} \sum_{k=1}^{nint} f(s_i, t_j, z_k) w_i w_j w_k \quad (7.6)$$

The exact integration of the element integrals is the basic idea in the finite element formulation. However, it will be obvious that we do only have an estimate of the degree of the polynomial to be integrated. However, it is a good estimate provided the shape of the element does not deviate too much from a square. This is what is called an ‘exact’ integration in the finite element context.

The discussion of the relation between Galerkin and Ritz method in section 15.2 in Appendix notes that it is possible to state how the finite element method converges when the number of degree of freedoms is increased. The solution converges from above down to the true minimum of some kind of function²⁹. However, this holds provided the integrals are integrated exactly. Further details about convergence of finite elements are discussed in chapter 10. The use of numerical integrations gives a possibility to improve the finite element methods. Underintegration may make it possible to converge faster, but not necessarily from above. This is illustrated in Figure 7.1. The numerical integration is sometimes reduced on some terms of the matrices in order to improve elements. The technique is called selective reduced integration, see section 9.2.

Thus the numerical integration process is not only efficient but can also be used as a ‘trick’ to improve elements in different respects. They can be underintegrated or selective underintegrated when only some specific terms of the integrals are underintegrated. However, too few integration points may lead to failure as it may occur that the contribution from certain combination of element variables may be lacking and lead to a matrix with a too high condition number. In terms of mechanical problems it is called zero-energy deformation. The sampling of strain and stress that occurs at the integration points do not discover any strain due to these deformation modes. This must be avoided or prevented.

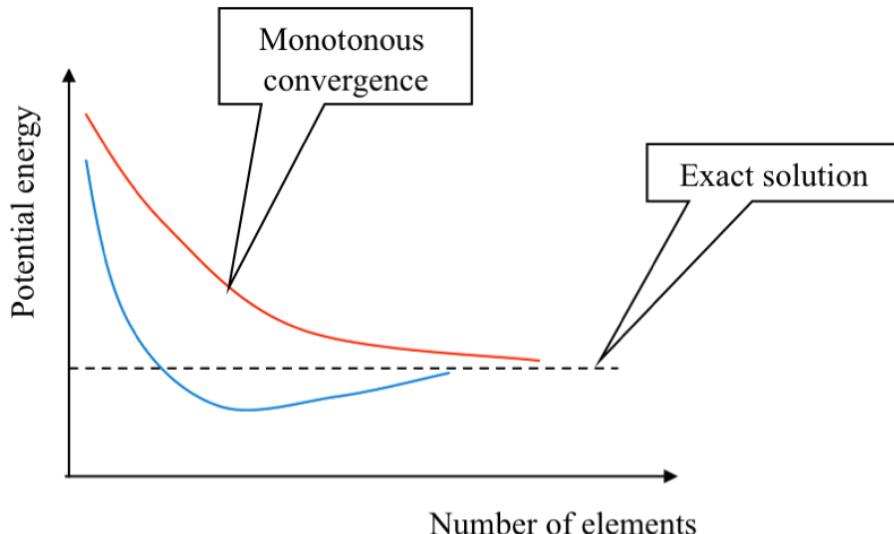


Figure 7.1. Convergence in terms of total potential energy in case of elastic problem.

²⁹ It is the total potential energy in case of elasticity problems.

8 Axisymmetric problems

The previous studied 2nd order equation that is applicable to various physical problems becomes somewhat more involved for axisymmetric problems. We will still only need one coordinate, the radius r , as we assume axisymmetric geometry and loading so that $u(r)$. We will illustrate this by writing Eq. (6.64) for axisymmetric problems. The loading will not be gravity in the following case but a centripetal force. Thus we make once more a mechanical interpretation of the equation as the equilibrium equation for a circular plate that is rotating with the angular speed ω around its centre axis. If the rotating plate is thin, then it is called plane stress³⁰ as the stress is zero in the thickness direction. If it is thick, then it is called plane strain as the strain is zero in the thickness direction. Although there is only one displacement, there are two stress and strain components. One component is in the radial direction and the other is in the hoop or circumferential direction. The coupling between the stress in r -direction and strain in φ -direction (and vice-versa) is the Poisson's ratio³¹, ν , seen in the constitutive relation in Figure 8.1. The figure shows the fundamental governing equations. Eliminating stress and strain in the equilibrium equation

$$L(u) = \frac{d\sigma_r}{dr} + \frac{1}{r}(\sigma_r - \sigma_\varphi) + f_r = 0 \quad \text{where } r \in [R_i, R_y], \quad (8.1)$$

gives the Navier's equation

$$L(u) = \frac{d^2u}{dr^2} + \frac{1}{r} \frac{du}{dr} - \frac{u}{r^2} + \frac{1-\nu^2}{E} f_r = 0 \quad \text{where } r \in [R_i, R_y], \quad (8.2)$$

where E is the elastic modulus and ν is Poisson's ratio. f_r is the volumetric loading, which in the case of a rotating disc is

$$f_r = \rho r \omega^2. \quad (8.3)$$

ω is the angular velocity and ρ is the density. The boundary conditions are applied at the inner, R_i , and outer, R_y , radii. The boundary conditions will be taken as

$$B_n(R_i) = \sigma_r(R_i) = 0 \quad \text{and} \quad (8.4)$$

$$B_n(R_y) = \sigma_r(R_y) = 0. \quad (8.5)$$

Hooke's law giving stresses from strains is

$$\begin{bmatrix} \sigma_r \\ \sigma_\varphi \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \end{bmatrix} \quad (8.6)$$

and the strain-displacement relations are

$$\begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \end{bmatrix} = \begin{bmatrix} \frac{du}{dr} \\ \frac{u}{r} \end{bmatrix} - \quad (8.7)$$

It turns out that the analytic solution of this problem is static determined, i.e. the stress distribution can be directly solved from Eq. (8.1) if only internal and external stresses are

³⁰ http://en.wikipedia.org/wiki/Plane_stress

³¹ http://en.wikipedia.org/wiki/Poisson%27s_ratio

prescribed. Therefor no influence of material properties is seen in the analytic solution for the stresses below if the centripetal loading is excluded. The solution is of the format

$$\sigma_r = A - \frac{B}{r^2} - \frac{3+\nu}{8} \rho \omega^2 r^2 \quad (8.8)$$

and

$$\sigma_\varphi = A + \frac{B}{r^2} - \frac{1+3\nu}{8} \rho \omega^2 r^2. \quad (8.9)$$

The boundary conditions in Eq.s (8.4) and (8.5) gives

$$A = \frac{3+\nu}{8} \rho \omega^2 \frac{(R_i^2 R_y^2 - R_i^2)}{R_i^2} \text{ and} \quad (8.10)$$

$$B = \frac{3+\nu}{8} \rho \omega^2 R_i^2 R_y^2 \text{ and} \quad (8.11)$$

Leading to

$$\sigma_r = \frac{3+\nu}{8} \rho \omega^2 \left[\frac{(R_i^2 R_y^2 - R_i^2)}{R_i^2} - \frac{R_i^2 R_y^2}{r^2} - r^2 \right]. \quad (8.12)$$

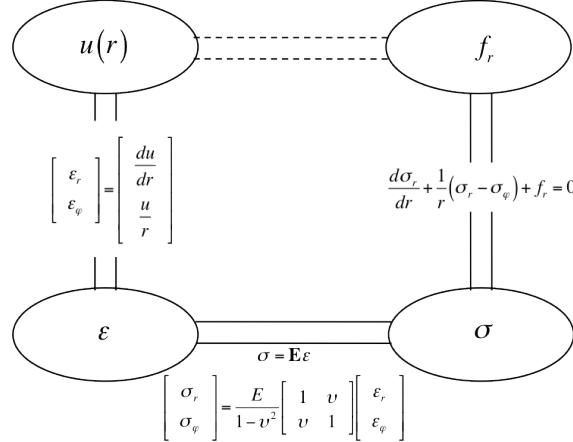


Figure 8.1. Tonti diagram for plane stress relations in linear, elastic problems.

The formulation element stiffness matrix and consistent load vector for Eq. (8.2) can be pursued in various ways. It is simplest for all variants to assume that we have one element over the whole domain. We can, see section 15.2 in appendix;

1. Use Ritz method based on minimisation of a functional, it is the total potential energy in the current context. This formulation is symmetric at the outset. However, it is limited to elastic problems.
2. The other alternative is to use a Galerkin approach with partial integration giving same derivatives on weight and trial functions giving a symmetric stiffness matrix. This was introduced in chapter 6 and pursued in section 6.2 for a rod element. This corresponds to the principle of virtual power for mechanical problems. We will use this approach current context.

The principle of virtual power states that the sum of internal and external virtual power or work³² is zero at equilibrium. This can be written as

³² http://en.wikipedia.org/wiki/Virtual_work

$$\delta W_{\text{int}} = \delta W_{\text{ext}} . \quad (8.13)$$

This will be written as

$$\int_{R_i}^{R_y} \delta \varepsilon^T \sigma \cdot 2\pi r dr = \int_{R_i}^{R_y} \delta u^T f_r \cdot 2\pi r dr \quad (8.14)$$

for the current problem. The factor 2π is usually ignored as it cancels. Notice the factor $2\pi r dr$ as we integrate over a ring with thickness dr and diameter $2\pi r$. δu is an admissible small virtual variation of the displacement field, the virtual displacement field. It is a purely mathematical device used to derive our approximate solution. We will later point out that it serves the same purpose as the weight functions in the Galerkin approach. The term admissible means that u must fulfil essential boundary conditions and that the correct relations are used to obtain $\delta \varepsilon$ from δu . Thus we have $\delta u = 0$ wherever we have a prescribed displacement. It is the same requirement for the Ritz method as noted in section 15.2 in appendix.

Eq. (8.14) becomes

$$\int_{R_i}^{R_y} \begin{bmatrix} \delta \varepsilon_r & \delta \varepsilon_\varphi \end{bmatrix} \begin{bmatrix} \sigma_r \\ \sigma_\varphi \end{bmatrix} \cdot r dr = \int_{R_i}^{R_y} \delta u^e(x) f_r \cdot r dr \quad (8.15)$$

The interpolation of the displacement over the element is written as

$$u^e(r) = \mathbf{N}\mathbf{u} = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \quad (8.16)$$

Thus we use a two node element and the shape functions are the same as in section 6.4. Eq. (8.7) gives

$$\varepsilon = \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \end{bmatrix} = \mathbf{B}\mathbf{u} = \begin{bmatrix} \frac{dN_1}{dr} & \frac{dN_2}{dr} \\ \frac{N_1}{r} & \frac{N_2}{r} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (8.17)$$

Notice the appearance of $1/r$ -terms. The matrix will be evaluated at integration points in the interior of the elements as discussed in chapter 7. Thus r will never be zero. Eq. (8.16) will also hold for the virtual nodal displacement and virtual displacement field

$$\delta u^e = \mathbf{N} \delta \mathbf{u} . \quad (8.18)$$

The strain-displacement relation will connect virtual displacements to virtual strains

$$\delta \varepsilon = \mathbf{B} \delta \mathbf{u} . \quad (8.19)$$

Inserting Eq.s (8.18) and (8.19) into Eq. (8.15) gives

$$\int_{R_i}^{R_y} \left(\mathbf{B} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \right)^T \begin{bmatrix} \sigma_r \\ \sigma_\varphi \end{bmatrix} \cdot r dr = \int_{R_i}^{R_y} \left(\mathbf{N} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \right)^T f_r \cdot r dr \quad (8.20)$$

Furthermore we replace stresses with strains according to Eq. (8.6) and use Eq. (8.17) to replace the strains with nodal displacements. We also use $\mathbf{E}^T = \mathbf{E}$. Then Eq. (8.18) becomes

$$\delta \mathbf{u}^T \int_{R_i}^{R_y} \mathbf{B}^T \mathbf{E} \mathbf{B} \cdot r dr \cdot \mathbf{u} = \delta \mathbf{u}^T \int_{R_i}^{R_y} \mathbf{N}^T f_r \cdot r dr . \quad (8.21)$$

This should be valid for all values of the virtual displacements at the nodes. Thus we must have

$$\int_{R_i}^{R_y} \mathbf{B}^T \mathbf{E} \mathbf{B} \cdot r dr \cdot \mathbf{u} = \int_{R_i}^{R_y} \mathbf{N}^T f_r \cdot r dr \quad (8.22)$$

or shorter

$$\mathbf{K}\mathbf{u} = \mathbf{f}_{ext}. \quad (8.23)$$

Thus we have obtained the formula for element stiffness matrix

$$\mathbf{k} = \int_{I^e} \mathbf{B}^T \mathbf{E} \mathbf{B} \cdot r dr. \quad (8.24)$$

The consistent nodal loads in case of centripetal loading becomes

$$\mathbf{f}_{ext} = \rho \omega^2 \int_{I^e} \mathbf{N}^T r^2 dr. \quad (8.25)$$

The integrals above are somewhat more involved than before and numerical integration can be used although analytic integration is possible.

The principle of virtual work used above can be compared to the Galerkin approach used earlier. The case leading to Eq. (6.53) will be taken as comparison. This equation, if assuming that the whole domain consists of one element, becomes

$$\int_{x_1}^{x_2} \mathbf{B}^T \mathbf{B} dx \mathbf{u} = \int_{x_1}^{x_2} \mathbf{N}^T \mathbf{N} dx \mathbf{u} - \int_{x_1}^{x_2} \mathbf{N}^T \cdot \mathbf{1} dx. \quad (8.26)$$

Comparing the fundamental equation, Eq. (6.1), with the current Eq. (8.2) shows that the left hand side above corresponds to the same term as Eq. (8.24) and that the last term above corresponds to Eq. (8.25). The derivation based on the principle of virtual work will be identical to the Galerkin method if we set $\delta\mathbf{u} = \mathbf{1}$. However, the similarity does not show up until we have performed the partial integration in the Galerkin method. This started at Eq. (6.7) and was based on inclusion of the natural boundary condition into the residual, Eq. (6.6). Thus the principle of virtual work is a shorter route to the formulation. Naturally, the use of this short cut requires the knowledge of a similar principle for the equation in focus. The principle of virtual work is general applicable in mechanics, also for nonlinear mechanics.

Exercise 8.1 (Computational problem)

Solve the rotating disc problem described above. Find an approximate solution using the using a two node element. The basic equations are given above.

Implement the solution in Matlab™ and evaluate it with respect to the exact solution. Make the comparison both for the radial displacement u and the radial and hoop stresses. Perform a convergence study like, for example, in Table 6.5. Evaluate error in terms of the energy norm

$$\Pi = \int_{R_i}^{R_y} \frac{1}{2} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \cdot r dr - \int_{R_i}^{R_y} u^T f_r \cdot r dr \quad (8.27)$$

Thus insert the exact solution into Eq. (8.27) and integrate. This is the true minimum. Compute the total potential energy from the finite element solution for varying number of nodes. The equation above becomes

$$\Pi = \frac{1}{2} \mathbf{U}^T \mathbf{K} \mathbf{U} - \mathbf{U}^T \mathbf{F}_{ext}$$

in the finite element case. Thus after solving for all nodal displacements, \mathbf{U} , then performing the multiplications above with the global stiffness matrix, \mathbf{K} , and the load vector \mathbf{F}_{ext} , yields the approximate value.

9 Higher order differential equations

The previous discussions have been limited to second order equations. They can describe a large range of physical phenomena. This chapter goes one step further. We will formulate finite elements for a fourth order differential equation. We will now discuss more about continuity requirements on the shape functions. They are important for the convergence properties of the finite element method discussed in detail in chapter 10. The equation is written as

$$\frac{d^4w}{dx^4} - \frac{q(x)}{EI} = 0, \quad (9.1)$$

w is the bending of the centre axis of the beam and $q(x)$ is a distributed vertical load. E is modulus of elasticity and I is area moment of the cross-section. The product EI is sometimes called bending stiffness of the beam. This equation will be related to bending of beams according to the Bernoulli (engineering) beam theory³³. It is applicable to slender beams. The slenderness of a beam is the quotient between its height and length. Another theory for less slender beams, the Timoshenko beam theory³⁴, will also be investigated. The governing equations is then two coupled second order equation and we will use this theory to demonstrate selective reduced integration mentioned in chapter 7.

The Ritz approach will be used in the finite element formulation, see sections 3.2 and in the appendix, section 15.2. Ritz method corresponds in this context to minimisation of the total potential energy of the system³⁵. To be more correct, the functional need only obtain a stationary value. It can be a maximum for unstable solutions or a minimum for stable solutions. Usually, the latter are of interest and therefore search for the minimum of potential energy is more common. However, the conditions we will use, first derivative must be zero, is valid for both cases. The second derivatives determines whether it is a stable, for a negative second derivative, or unstable, a positive second derivative.

Displacements based finite elements are the dominating class of elements for mechanical problems. The deformations, displacements and rotations, are the primary variables. There are stress-based elements for some few applications (torsion). They are based on minimisation of what is called complementary potential energy. There are also elements that are a mixture between these two groups exists – mixed elements. Another types of elements are called hybrid elements. They have independent assumptions about the fields in their interior and their boundaries. These formulations are not discussed following.

9.1 Bernoulli beam

The theory for the bending deformation of the Bernoulli beam theory or sometimes called technical beam theory, applicable to slender beams is given in section 15.4.

The solution of beam problems should make the total potential energy, see Eq. (15.36), minimum. The approximate solution must fulfil essential boundary conditions, then the best possible solution is the one that minimises the functional

$$\Pi = \int_L \frac{1}{2} EI \left(\frac{d^2w}{dx^2} \right)^2 dx - \int_L q w dx. \quad (9.2)$$

³³ http://en.wikipedia.org/wiki/Euler%20%93Bernoulli_beam_theory

³⁴ http://en.wikipedia.org/wiki/Timoshenko_beam_theory

³⁵ http://en.wikipedia.org/wiki/Minimum_total_potential_energy_principle

Fulfilling essential boundary conditions are done by assigning appropriate nodal values to the model as will be seen in section 9.3. They are prescribed values for w as well as its first derivative w' . The latter is the slope or rotation of the beam. Therefore, the essential boundary conditions are sometimes called geometric boundary conditions.

The element stiffness matrix and load vector will be formulated by applying Eq. (9.2) to one element. The fundamental equilibrium equation expressed in displacement, Eq. (9.1), is a fourth order differential equation. However, the approximate solution must only have a non-zero second derivative w'' existing in Eq. (9.2). Thus it needs to be at least a cubic polynomial. Furthermore, the function must give continuous derivatives/slope between elements. Otherwise there is a risk that the split of the integral Eq. (9.2) into elements may lose energy in the boundaries between these elements. But if the slope is continuous between elements, then the curvature must at least be finite and then no energy is lost at the point joining two elements. The fulfilment of the required C^1 -continuity³⁶ of the deformation w between the elements and the essential boundary conditions is simplified by the use of displacement and slope/rotation as nodal variables.

We will simplify the derivation of the element matrix by assuming that the structure has only one element. The shape functions of this element are used to calculate the contribution to Eq. (9.2) and then the stationary value of the functional will give the wanted formulas for element stiffness matrix as well as nodal load vector due to the distributed load q .

The two node beam element with two degree of freedoms per element shown in Figure 9.1 will be formulated. Each node has a bending displacement as well as a rotation. This rotation is the slope of the beam is equal to the first derivative of w for small deformations. The four nodal values can define the four coefficients of a third order polynomial. The element field is written as

$$w(x) = \mathbf{N}\mathbf{u} = \begin{bmatrix} N_{w1} & N_{\theta 1} & N_{w2} & N_{\theta 2} \end{bmatrix} \mathbf{u}. \quad (9.3)$$

The shape functions are the polynomials shown below.

$$w(x) = \begin{bmatrix} 1 - 3\frac{x^2}{l^e} + 2\frac{x^3}{l^{e^3}} & -x + 2\frac{x^2}{l^e} - \frac{x^3}{l^{e^2}} & 3\frac{x^2}{l^{e^2}} - 2\frac{x^3}{l^{e^3}} & \frac{x^2}{l^e} - \frac{x^3}{l^{e^2}} \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix}. \quad (9.4)$$

These functions have the properties

$$\begin{aligned} N_{w1}(0) &= 1, N_{w1}(l^e) = 0 \\ N_{\theta 1}(0) &= 0, N_{\theta 1}(l^e) = 0 \\ N_{w2}(0) &= 0, N_{w2}(l^e) = 1 \\ N_{\theta 2}(0) &= 0, N_{\theta 2}(l^e) = 0 \end{aligned} \quad (9.5)$$

and

³⁶ The previous case fulfilled C^0 -continuity. The notation C^n -continuity of a function means that the n :th derivative of the function is continuous.

$$\begin{aligned}
\frac{dN_{w1}}{dx}(0) &= 0, \quad \frac{dN_{w1}}{dx}(l^e) = 0 \\
\frac{dN_{\theta1}}{dx}(0) &= -1, \quad \frac{dN_{\theta1}}{dx}(l^e) = 0 \\
\frac{dN_{w2}}{dx}(0) &= 0, \quad \frac{dN_{w2}}{dx}(l^e) = 0 \\
\frac{dN_{\theta2}}{dx}(0) &= 0, \quad \frac{dN_{\theta2}}{dx}(l^e) = -1
\end{aligned} \tag{9.6}$$

The cubic equation is the exact solution to a beam bending problem where only nodal loads are applied, see Appendix 15.4.

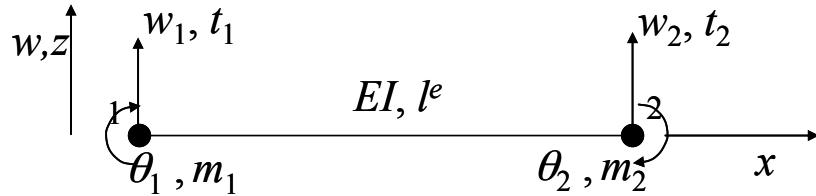


Figure 9.1. Bernoulli beam element.

We need the second derivatives, i.e. the curvature, of the beam in Eq. (9.2). This is

$$\kappa(x) = -\frac{d^2\mathbf{N}}{dx^2}\mathbf{u} = \mathbf{B}\mathbf{u} \tag{9.7}$$

with

$$\mathbf{B} = \left[\begin{array}{cccc} \frac{6}{l^{e2}} - 12\frac{x}{l^{e3}} & -\frac{4}{l^e} + 6\frac{x}{l^{e2}} & -\frac{6}{l^{e2}} + 12\frac{x}{l^{e3}} & -\frac{2}{l^e} + 6\frac{x}{l^{e2}} \end{array} \right]. \tag{9.8}$$

Then Eq. (9.2) is written as, assuming only one element in the model,

$$\Pi = \int_{l^e} \frac{1}{2} EI (\mathbf{B}\mathbf{u})^T \mathbf{B}\mathbf{u} dx - \int_{l^e} (\mathbf{N}\mathbf{u})^T q dx \tag{9.9}$$

The nodal variables, \mathbf{u} , does not depend on coordinates and can be moved outside the integrals. This gives

$$\Pi = \frac{1}{2} \mathbf{u}^T \int_{l^e} EI \mathbf{B}^T \mathbf{B} dx \mathbf{u} - \mathbf{u}^T \int_{l^e} \mathbf{N}^T q dx, \tag{9.10}$$

or shorter

$$\Pi = \frac{1}{2} \mathbf{u}^T \mathbf{k} \mathbf{u} - \mathbf{u}^T \mathbf{f}_{ext}. \tag{9.11}$$

The stationary value of this expression w.r.t. \mathbf{u} is

$$\frac{\partial \Pi}{\partial \mathbf{u}} = \mathbf{k} \mathbf{u} - \mathbf{f}_{ext} = \mathbf{0} \Leftrightarrow \mathbf{k} \mathbf{u} = \mathbf{f}_{ext}. \tag{9.12}$$

Now we can identify element stiffness matrix as

$$\mathbf{k} = \int_{l^e} EI \mathbf{B}^T \mathbf{B} dx \text{ and} \tag{9.13}$$

consistent nodal load vector due to the distributed load q

$$\mathbf{f}_{ext} = \int_{l^e} \mathbf{N}^T q dx. \tag{9.14}$$

The word consistent means that the nodal forces and moments in \mathbf{f}_{ext} give the same contribution to the potential of the loads as the original distributed load would have given in Eq. (9.2).

The stiffness matrix becomes

$$\mathbf{k} = \int_{l^e} EI \begin{bmatrix} \frac{6}{l^{e2}} - \frac{12x}{l^{e3}} \\ -\frac{4}{l^e} + \frac{6x}{l^{e2}} \\ -\frac{6}{l^{e2}} + \frac{12x}{l^{e3}} \\ -\frac{2}{l^e} + \frac{6x}{l^{e2}} \end{bmatrix} \begin{bmatrix} \frac{6}{l^{e2}} - \frac{12x}{l^{e3}} & -\frac{4}{l^e} + \frac{6x}{l^{e2}} & -\frac{6}{l^{e2}} + \frac{12x}{l^{e3}} & -\frac{2}{l^e} + \frac{6x}{l^{e2}} \end{bmatrix} dx. \quad (9.15)$$

That can be solved analytically or numerically. The latter would require, see chapter 7, two integration points to for this 2nd order polynomial. We would get

$$\mathbf{k} = \frac{EI}{l^{e3}} \begin{bmatrix} 12 & -6l^e & -12 & -6l^e \\ -6l^e & 4l^{e2} & 6l^e & 2l^{e2} \\ -12 & 6l^e & 12 & 6l^e \\ -6l^e & 2l^{e2} & 6l^e & 4l^{e2} \end{bmatrix} \quad (9.16)$$

and for the load, assuming $q=\text{constant}$ along the beam

$$\mathbf{f}_{ext} = \begin{bmatrix} t_1 \\ m_1 \\ t_2 \\ m_2 \end{bmatrix} = \int_{l^e} \begin{bmatrix} 1 - 3\frac{x^2}{l^{e2}} + 2\frac{x^3}{l^{e3}} \\ -x + 2\frac{x^2}{l^e} - \frac{x^3}{l^{e2}} \\ 3\frac{x^2}{l^{e2}} - 2\frac{x^3}{l^{e3}} \\ \frac{x^2}{l^e} - \frac{x^3}{l^{e2}} \end{bmatrix} q dx = \begin{bmatrix} \frac{1}{2} \\ -\frac{l^e}{12} \\ \frac{1}{2} \\ \frac{l^e}{12} \end{bmatrix} ql^e. \quad (9.17)$$

One can check with elementary cases for a beam that the above edge loads on a beam gives the same displacements and rotations as the distributed load would have done. However, the bending between the nodes will be different.

The same matrix can be obtained by using elementary cases for a beam loaded with unit forces and/or moments at the nodes as this is the exact solution for a beam with constant properties along its length and only nodal loads. Why?

An analysis of a beam problem using this element is shown in section 9.3.

9.2 Timoshenko beam

The Timoshenko beam element in Figure 9.2 is similar to the previous Bernoulli beam but requires some additional material properties. The equilibrium between bending moment and shear leads to

$$\frac{d^3\theta}{dx^3} - \frac{q(x)}{EI} = 0. \quad (9.18)$$

where θ is the rotation of the cross-section of the beam. It is related to the slope of the beam due to shearing caused by shear stresses. The coupling equation is

$$\frac{dw}{dx} = \theta - \frac{\alpha}{AG} EI \frac{d^2\theta}{dx^2}. \quad (9.19)$$

Thus there are two coupled variables for the solution of the Timoshenko beam theory. The classical theory in Eq. (9.1) is obtained when $\frac{dw}{dx} = \theta$. The solution approaches this value when the beam becomes more slender. See appendix, section 15.5, for more about the Timoshenko beam theory.

The Ritz method will be used to obtain the finite element formulation. The total potential energy³⁷ to be minimised, see section 15.5, is

$$\Pi = \int_L \frac{1}{2} \begin{bmatrix} \frac{d\theta}{dx} & \gamma \end{bmatrix} \begin{bmatrix} EI & 0 \\ 0 & \frac{GA}{\alpha} \end{bmatrix} \begin{bmatrix} \frac{d\theta}{dx} \\ \gamma \end{bmatrix} dx - \int_L qwdx. \quad (9.20)$$

We will assume that the whole problem consists of one element and then we can determine the stationary value of this function versus the nodal variables of the element. We will base the formulation on the element in Figure 9.2. It looks like the beam in Figure 9.1 but additional material properties are needed. They are related to the shear of the beam cross-section. G is the shear modulus and

$$A_s = \frac{A}{\alpha} \quad (9.21)$$

depends on the shape of the cross-section.

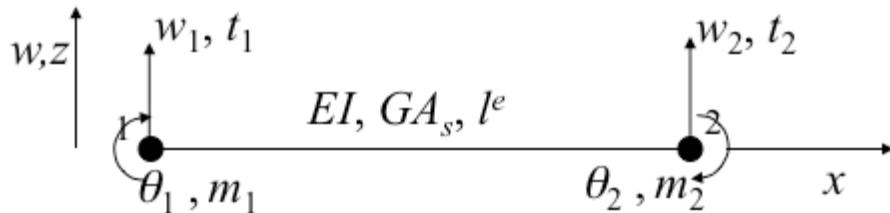


Figure 9.2. Two node Timoshenko beam element.

We use an isoparametric approach by

$$x(s) = N_1 x_1 + N_2 x_2 \quad (9.22)$$

with

$$N_1 = \frac{1}{2}(1-s) \quad N_2 = \frac{1}{2}(1+s). \quad (9.23)$$

The rotation and displacements are now interpolated independently as

$$\mathbf{u}(s) = \begin{bmatrix} w \\ \theta \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{N}\mathbf{u}. \quad (9.24)$$

³⁷ The stored energy can also be written as $\int_L \frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 + \frac{1}{2} \frac{GA}{\alpha} \gamma^2 dx$ showing that the energies can be added separately.

This is a large difference to the Bernoulli beam case in Eq. (9.4) where we have

$$\frac{dw}{dx} = \theta. \quad (9.25)$$

The elastic energy, first term in Eq. (9.26), requires

$$\boldsymbol{\epsilon} = \begin{bmatrix} d\theta \\ dx \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 & \frac{dN_1}{dx} & 0 & \frac{dN_2}{dx} \\ \frac{dN_1}{dx} & N_1 & \frac{dN_2}{dx} & N_2 \\ \gamma & & & \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{B}\mathbf{u}. \quad (9.26)$$

We can compute local derivatives, with respect to the coordinate s . However, the global derivatives are needed. The isoparametric mapping is the same as in Eq.s (6.55). The Jacobian of the mapping is

$$J \frac{dx}{ds} = \frac{l^e}{2} \quad (9.27)$$

where l^e is the length of the element. The \mathbf{B} -matrix becomes

$$\mathbf{B} = \begin{bmatrix} 0 & -\frac{1}{l^e} & 0 & \frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{2}(1-s) & \frac{1}{l^e} & \frac{1}{2}(1+s) \end{bmatrix}. \quad (9.28)$$

The total potential energy³⁸, section 15.5, is

$$\Pi = \int_L^l \frac{1}{2} \left[\frac{d\theta}{dx} \quad \gamma \right] \begin{bmatrix} EI & 0 \\ 0 & \frac{GA}{\alpha} \end{bmatrix} \begin{bmatrix} d\theta \\ \gamma \end{bmatrix} dx - \int_L^l qwdx. \quad (9.29)$$

The elastic stored energy in the first term is leads to the stiffness matrix in the same way as in for the Bernoulli beam. The element stiffness matrix is

$$\mathbf{k} = \int_{l^e}^l \mathbf{B}^T \mathbf{E} \mathbf{B} dx \quad (9.30)$$

or

$$\mathbf{k} = \int_{-1}^1 \begin{bmatrix} 0 & -\frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{2}(1-s) \\ 0 & \frac{1}{l^e} \\ \frac{1}{l^e} & \frac{1}{2}(1+s) \end{bmatrix} \begin{bmatrix} EI & 0 \\ 0 & GA_s \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{l^e} & 0 & \frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{2}(1-s) & \frac{1}{l^e} & \frac{1}{2}(1+s) \end{bmatrix} \frac{l^e}{2} ds. \quad (9.31)$$

³⁸ The stored energy can also be written as $\int_L^l \frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 + \frac{1}{2} \frac{GA}{\alpha} \gamma^2 dx$ showing that the energies can be added separately.

Multiplying the matrices gives

$$\mathbf{k} = \int_{-1}^1 \begin{bmatrix} \frac{GA_s}{l^2} & \frac{-GA_s}{2l^e}(1-s) & \frac{-GA_s}{l^2} & \frac{-GA_s}{2l^e}(1+s) \\ \frac{-GA_s}{2l^e}(1-s) & \frac{EI}{l^2} + \frac{GA_s}{4}(1-s)^2 & \frac{GA_s}{2l^e}(1-s) & \frac{-EI}{l^2} + \frac{GA_s}{4}(1-s^2) \\ \frac{-GA_s}{l^2} & \frac{GA_s}{2l^e}(1-s) & \frac{GA_s}{l^2} & \frac{GA_s}{2l^e}(1+s) \\ \frac{-GA_s}{2l^e}(1+s) & \frac{-EI}{l^2} + \frac{GA_s}{4}(1-s^2) & \frac{GA_s}{2l^e}(1+s) & \frac{EI}{l^2} + \frac{GA_s}{4}(1+s)^2 \end{bmatrix} \frac{l^e}{2} ds. \quad (9.32)$$

A split is done before solving the integral above. The ‘stress’-‘strain’ relation matrix is diagonal as the strain energies due to shear and normal straining are uncoupled. The stiffness matrix is related to the stored elastic energy as in Eq. (9.11) and can therefore also be uncoupled³⁸. It is written as

$$\mathbf{k} = \mathbf{k}_b + \mathbf{k}_s \quad (9.33)$$

where

$$\mathbf{k}_b = \int_{-1}^1 \frac{EI}{l^{e2}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \frac{l^e}{2} ds = \frac{EI}{l^e} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (9.34)$$

and

$$\mathbf{k}_s = \int_{-1}^1 \frac{GA_s}{l^{e2}} \begin{bmatrix} 1 & -\frac{l^e}{2}(1-s) & -1 & -\frac{l^e}{2}(1+s) \\ -\frac{l^e}{2}(1-s) & \left(\frac{l^e}{2}\right)^2(1-s)^2 & \frac{l^e}{2}(1-s) & \left(\frac{l^e}{2}\right)^2(1-s^2) \\ -1 & \frac{l^e}{2}(1-s) & 1 & \frac{l^e}{2}(1+s) \\ -\frac{l^e}{2}(1+s) & \left(\frac{l^e}{2}\right)^2(1-s^2) & \frac{l^e}{2}(1+s) & \left(\frac{l^e}{2}\right)^2(1+s)^2 \end{bmatrix} \frac{l^e}{2} ds. \quad (9.35)$$

Exact integration gives for the latter gives

$$\mathbf{k}_s = \frac{GA_s}{l^e} \begin{bmatrix} 1 & -l^e/2 & -1 & -l^e/2 \\ -l^e/2 & l^{e2}/3 & l^e/2 & l^{e2}/6 \\ -1 & l^e/2 & 1 & l^e/2 \\ -l^e/2 & l^{e2}/6 & l^e/2 & l^{e2}/3 \end{bmatrix}. \quad (9.36)$$

We will in the next chapter apply this exactly integrated element stiffness matrix and find problem. We will then also use a version where the highest order terms in Eq. (9.35) are underintegrated. We use one integration point formula according to Table 7.1 and get

$$\mathbf{k}_s(2,2) = \frac{GA_s l^e}{8l^e} \int_{-1}^1 (1-s)^2 ds = \frac{GA_s l^e}{8} (1-0)^2 \cdot 2 = \frac{GA_s l^e}{4} \quad (9.37)$$

and the same value for $\mathbf{k}_s(4,4)$. The other term to be approximated is

$$\mathbf{k}_s(2,4) = \mathbf{k}_s(4,2) = \frac{GA_s l^e}{8l^e} \int_{-1}^1 (1-s^2) ds = \frac{GA_s l^e}{8} (1-0)^2 \cdot 2 = \frac{GA_s l^e}{4}. \quad (9.38)$$

Then we have

$$\tilde{\mathbf{k}}_s = \frac{GA_s}{l^e} \begin{bmatrix} 1 & -l^e/2 & -1 & -l^e/2 \\ -l^e/2 & l^{e^2}/4 & l^e/2 & l^{e^2}/4 \\ -1 & l^e/2 & 1 & l^e/2 \\ -l^e/2 & l^{e^2}/4 & l^e/2 & l^{e^2}/4 \end{bmatrix}. \quad (9.39)$$

The consistent nodal load vector due to the distributed load \mathbf{q} is

$$\mathbf{f}_{ext} = \int_{-1}^1 \frac{1}{2} \begin{bmatrix} (1-s) & 0 \\ 0 & (1-s) \\ (1+s) & 0 \\ 0 & (1+s) \end{bmatrix} \begin{bmatrix} q \\ m \end{bmatrix} \frac{l^e}{2} ds. \quad (9.40)$$

Notice the extension of \mathbf{q} to a vector with two parts. One is related to the deflection of the beam and the other to its rotation. Originally, Eq. (9.40) comes from the potential of the loads that is the loads multiplying the deformations in Eq. (9.29). Then we had only qw in the product. Now we have, see Eq. (9.24), deflection and rotation as fields describing the deformation independently. The potential of the loads is therefore both force multiplying deflection w and distributed moment multiplying rotations. Thus motivating Eq. (9.40).

Integration of above for constant distributed loads gives simply

$$\mathbf{f}_{ext} = \begin{bmatrix} \frac{ql^e}{2} \\ \frac{ml^e}{2} \\ \frac{2}{2} \\ \frac{ql^e}{2} \\ \frac{2}{2} \\ \frac{ml^e}{2} \end{bmatrix}. \quad (9.41)$$

9.3 Cantilever beam problem

We will compare finite element solutions with the analytic solution according to Bernoulli beam theory in section 15.4. A model with one element will be used to solve the cantilever beam with a point load, upper part in Figure 9.3. The reaction force and moment are replacing the wall that was drawn in Figure 15.5. They are unknown and their magnitudes comes from the fixed displacement and rotation at the wall, $x=0$.

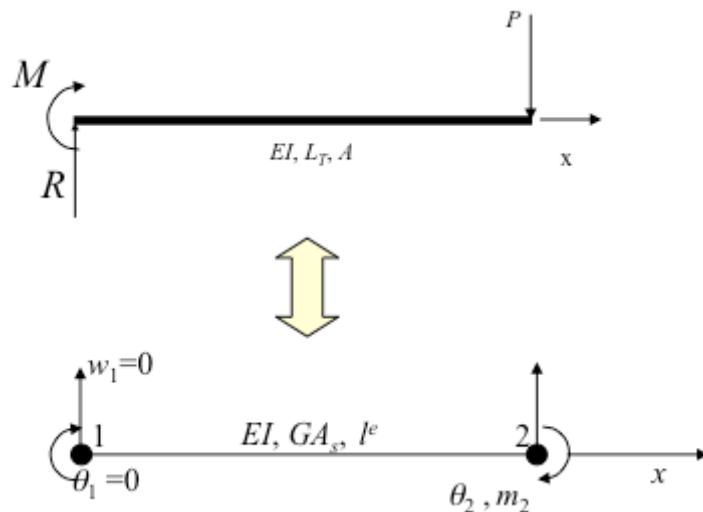


Figure 9.3. Cantilever beam with point load and a one element model.

The problem is shown and solved analytically in section 15.4 for the Bernoulli beam theory. The beam has a square cross-section with width B and height H . Thus we have

$$I = \frac{BH^3}{12}. \quad (9.42)$$

and

$$\alpha = \frac{10(1+\nu)}{12+11\nu}. \quad (9.43)$$

The latter gives

$$A_s = \frac{BH}{\alpha} = \frac{12+11\nu}{10(1+\nu)} BH. \quad (9.44)$$

The Bernoulli beam theory gives

$$w(x) = -\frac{PL}{2EI} x^2 + \frac{P}{6EI} x^3 = \frac{PL^3}{EI} \left(\frac{1}{6} \left(\frac{x}{L} \right)^3 - \frac{1}{2} \left(\frac{x}{L} \right)^2 \right). \quad (9.45)$$

The maximum displacement (downwards) is

$$w = -\frac{PL^3}{3EI}. \quad (9.46)$$

The maximum deflection according to the Timoshenko beam theory is

$$w_T = \frac{PL^3}{EI} \frac{1}{3} \left(1 + \frac{EI}{\alpha G A L^2} \right) = \frac{PL^3}{EI} \frac{1}{3} \left(1 + \frac{(12+11\nu)}{20} \left(\frac{H}{L} \right)^2 \right). \quad (9.47)$$

We used

$$G = \frac{E}{2(1+\nu)} = \frac{E}{2(1+0.25)} \quad (9.48)$$

for obtaining the last equality in Eq. (9.47).

The one element model is shown in the lower part of Figure 9.3. The fulfilment of the essential boundary conditions at $x=0$ requires the rotation and displacement at left end to be zero. The corresponding unknown reaction force and moment are included in the systems of equations below.

The Bernoulli beam gives, Eq. (8.14),

$$\frac{EI}{L^3} \begin{bmatrix} 12 & -6L & -12 & -6L \\ -6L & 4L^2 & 6L & 2L^2 \\ -12 & 6L & 12 & 6L \\ -6L & 2L^2 & 6L & 4L^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} R \\ M \\ -P \\ 0 \end{bmatrix}$$

The unknown displacement and rotation at the free end can be obtained from the system of equations above but first the essential boundary conditions must be inserted, the two zeros in the vector with unknown nodal degree of freedoms. This is only necessary to use the two last equations for our wanted deformation. The first two can be used afterwards to determine reaction force and moment at the left end. Thus the system to be solved is

$$\frac{EI}{L^3} \begin{bmatrix} 12 & 6L \\ 6L & 4L^2 \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \quad (9.49)$$

giving

$$\begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \frac{L}{12EI} \begin{bmatrix} 4L^2 & -6L \\ -6L & 12 \end{bmatrix} \begin{bmatrix} -P \\ 0 \end{bmatrix} = \frac{PL^3}{EI} \begin{bmatrix} -1 \\ 3 \\ 1 \\ 2 \end{bmatrix} \quad (9.50)$$

This is the exact solution as expected as our element has cubic variation in the displacement, which is sufficient to represent the theoretical solution. Particularly we check the deflection at the end with the analytic solution in Eq.

$$w_{LBB} = -\frac{PL^3}{3EI^3} \left[1 + \left(\frac{H}{L} \right)^2 \frac{(12+11\nu)}{20} \right] \quad (15.58)$$

We get

$$w_2 = -\frac{PL^3}{3EI} = w_{BBT} \quad (9.51)$$

The fully integrated Timoshenko beam, Eq. (9.34) and Eq. (9.36), gives,

$$\begin{bmatrix} EI \\ L \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} + \frac{GA_s}{L} \begin{bmatrix} 1 & -L/2 & -1 & -L/2 \\ -L/2 & L^2/3 & L/2 & L^2/6 \\ -1 & L/2 & 1 & L/2 \\ -L/2 & L^2/6 & L/2 & L^2/3 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} R \\ M \\ -P \\ 0 \end{bmatrix}$$

This leads to

$$\begin{bmatrix} GA_s \\ L \\ \frac{GA_s}{2} \\ \frac{GA_s}{2} + \frac{GA_s L}{3} \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \quad (9.52)$$

A ratio between shear and bending is introduced order to simplify the expression above

$$f_s = GA_s L \frac{L}{EI} \quad (9.53)$$

Eq. (8.34) can be written as

$$\frac{EI}{L} \begin{bmatrix} \frac{f_s}{L^2} & \frac{f_s}{2L} \\ \frac{f_s}{2L} & 1 + \frac{f_s}{3} \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \quad (9.54)$$

The algebraic solution is

$$\begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \frac{L}{EI \left(\frac{f_s}{L^2} \left(1 + \frac{f_s}{3} \right) - \left(\frac{f_s}{2L} \right)^2 \right)} \begin{bmatrix} \left(1 + \frac{f_s}{3} \right) & -\frac{f_s}{2L} \\ -\frac{f_s}{2L} & \frac{f_s}{L^2} \end{bmatrix} \begin{bmatrix} -P \\ 0 \end{bmatrix}$$

giving

$$w_2 = -\frac{PL^3}{EI} \frac{\left(\frac{1}{f_s} + \frac{1}{3} \right)}{\left(\frac{1}{f_s} + \frac{1}{12} \right)} \quad (9.55)$$

Assuming a square cross-section of the beam gives

$$I = \frac{BH^3}{12}, A = BH, \alpha = \frac{5}{6}, G = \frac{E}{2(1+\nu)} \quad (9.56)$$

Insertion of above into Eq. (9.53) gives

$$f_s = \frac{1}{6(1+\nu)} \frac{12BHL^2}{BH^3} \frac{5}{6} = \frac{1}{2(1+\nu)} \left(\frac{L}{H}\right)^2 \approx \left(\frac{L}{H}\right)^2 \quad (9.57)$$

This makes the deflection of the edge of the beam, Eq.(9.55),

$$w_2 = -\frac{PL^3}{EI} \frac{\left(2(1+\nu)\left(\frac{H}{L}\right)^2 + \frac{1}{3}\right)}{\left(2(1+\nu)\left(\frac{H}{L}\right)^2 + \frac{1}{12}\right)} \approx \left[\text{if } \frac{H}{L} \ll 1 \right] = -\frac{PL^3}{4EI} = 0.75w_{BBT} = 0.75w_{TBT}$$

However, there is a finite precision in the computer when solving the system of equations and for slender beams ($H \ll L$) then

$$1 + \frac{f_s}{3} \approx \frac{f_s}{3}$$

Then the computer will experience Eq. (9.54) as

$$\frac{EI}{L} \begin{bmatrix} \frac{f_s}{L^2} & \frac{f_s}{2L} \\ \frac{f_s}{2L} & \frac{f_s}{3} \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \quad (9.58)$$

This gives the solution

$$\begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} \approx -\frac{PL}{EI} \frac{1}{\frac{f_s}{12L^2}} \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{2L} \end{bmatrix}$$

Leading to

$$w_2 = -\frac{PL}{EI} \frac{12}{\left(\frac{f_s}{4L^2}\right)} = -\frac{PL^3}{3EI} \frac{12}{f_s}$$

This will be a very small number! Thus the solution is very bad. Before discussing this phenomenon, lets evaluate the results from the underintegrated Timoshenko beam, Eq. (9.39). That formulation gives

$$\frac{EI}{L} \begin{bmatrix} \frac{f_s}{L^2} & \frac{f_s}{2L} \\ \frac{f_s}{2L} & 1 + \frac{f_s}{4} \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \quad (9.59)$$

The solution becomes

$$\begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \frac{L}{EI \left(\frac{f_s}{L^2} \left(1 + \frac{f_s}{4} \right) - \left(\frac{f_s}{2L} \right)^2 \right)} \begin{bmatrix} \left(1 + \frac{f_s}{4}\right) & -\frac{f_s}{2L} \\ -\frac{f_s}{2L} & \frac{f_s}{L^2} \end{bmatrix} \begin{bmatrix} -P \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \frac{L}{EI \frac{f_s}{L^2}} \begin{bmatrix} \left(1 + \frac{f_s}{4}\right) & -\frac{f_s}{2L} \\ -\frac{f_s}{2L} & \frac{f_s}{L^2} \end{bmatrix} \begin{bmatrix} -P \\ 0 \end{bmatrix} = \frac{-PL^3}{EI f_s} \begin{bmatrix} 1 + \frac{f_s}{4} \\ -\frac{f_s}{2L} \end{bmatrix}$$

leading to

$$w_2 = -\frac{PL^3}{EI} \left(2(1+\nu) \left(\frac{H}{L} \right)^2 + \frac{1}{4} \right) \approx \left[\text{if } \frac{H}{L} \ll 1 \right] = -\frac{PL^3}{4EI} = 0.75w_L \quad (9.60)$$

This result will not suffer from any truncation error as the fully integrated version does. The underintegrated Timoshenko beam element handles the slender beam case much better than the fully integrated version. The table below summarise the result for the two Timoshenko element formulations and the Bernoulli beam element. The Timoshenko beam element has a linear interpolation of the rotation and the deflection. The deformation of a slender beam, i.e. a case where the Bernoulli beam theory approximation is good, is such that the rotation is the first derivative of the deflection. This cannot be handled well when the two fields are both linear polynomials. The fully integrated element ‘locks’ in order to resolved this issue. Thus the deflection and rotation becomes very small in an attempt to remove the shearing from the deformation of the beam. The underintegration is performed on the terms that have to do with the rotation field. Thus the element will, despite the use of a linear shape function, only experience one value for the rotation as this is sampled at the centre of the element. Therefore, this element performs better.

The above explanation of the phenomenon underlying the truncation problem of the fully integrated element can also be discussed in terms of energy. The fully integrated element will lock in order to reduce the strain energy due to the shearing. The underintegration of this part of the element stiffness matrix makes the element lose some of that strain energy and thereby softens the element.

Table 9.1. Computed maximum deflection of cantilever beam. Finite element results are normalised versus theoretical result for Bernoulli beam theory as the beam has a thickness/length ratio of 0.01.

Number of elements	Normalised results		
	Bernoulli beam element	Underintegrated Timoshenko beam element	Fully integrated Timoshenko beam element
1	0.9999	0.750	0.0003
2	0.9999	0.938	0.0012
4	0.9999	0.984	0.0049
8	0.9999	0.9961	0.0192
16	0.9999	0.9990	0.0726

10 Convergence of finite element formulations

The overall important requirement on a finite element is that it must converge to the exact solution when the mesh is refined. Naturally, this will not be possible for point loads and sharp corners where the theoretical solution gives a singularity. This means that the stress will at these points go towards infinity. The requirements on the approximate solutions can be summarised as Completeness and Continuity.

The shape functions must be taken from a complete set of functions. Typically, polynomial functions are used. Then we must start with the lowest terms in a series. This is in one dimension $c_0 + c_1x + c_2x^2 + \dots$ and in two dimensions $a_0 + a_1x + a_2y + a_3y^2 + a_4x^2 + a_5xy\dots$ and similar in three dimensions. This is visualised for two dimensions in Figure 10.1. The different element types (3 node triangle etc) are discussed in chapters **Error! Reference source not found.** and 12. If the formulation skips a lower order term, then convergence will not be achieved.

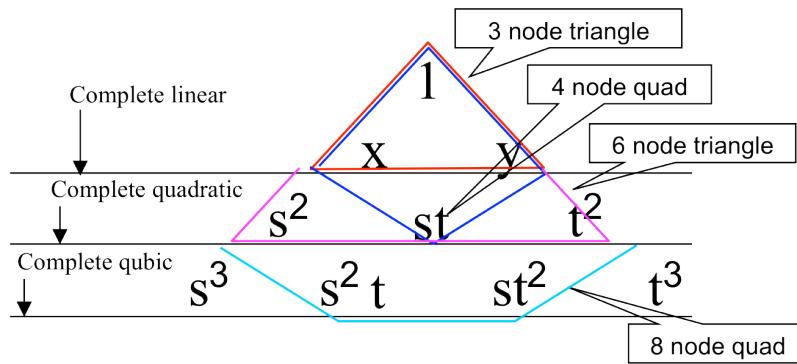


Figure 10.1. Pascal's triangle.

Furthermore, the field of an element must be able to represent a constant field leading to zero gradient. This is called rigid body motion in case of mechanical problems. This kind of motions includes no straining and thus no stresses should exist then. The two node element has no rigid body motion but the rod in section 6.5 has one rigid body displacement. It is when $u_1=u_2$ for an element. Eq. (6.70) gives the element nodal forces for such a displacement. Assume $a=u_1=u_2$. It gives

$$\frac{EA}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} = \frac{EA}{l^e} \begin{bmatrix} a-a \\ -a+a \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (10.1)$$

which is correct. However, a zero force vector must not be obtained for any other motion. This kind of deformation is called a zero-energy deformation or mechanism of an element. This may be the case for a correct element formulation but if too few numerical integration points are used to solve the element integral for the stiffness matrix.

The assumed field must naturally also be able to represent the derivatives existing in the formula for the element matrix. The finite element formulation has integrals over the whole domain of the problem. They are split into element integrals and then a minimum continuity is required between elements also. The continuity requirement can be summarised as the following. If $2m$ is the highest derivative in the fundamental equation, then the final integral for the element matrix has m as its highest derivative. Then the $(m-1)$:th derivative of the shape functions must exist. Furthermore, the derivative must be continuous between elements. We typically have $m=2$ and then need C^0 -continuous shape functions between elements. However, they must be C^1 -continuous for the beam in Eq. (9.1) with the element integral in Eq. (9.2).

11 Isoparametric mapping in two dimensions

The concept of isoparametric mapping was introduced in section 6.4 in the context of one-dimensional elements. However, it was unnecessary for this simple type of elements as the element integrals always have simple boundaries, left and right end of element. The motivation for showing the formulation was that is simple to show in this context. It was a prelude to chapters **Error! Reference source not found.** and 12 where the isoparametric mapping is a necessary step in the finite element formulation. There it is shown for a two-dimensional element.

Chapter 0 describe the numerical integration technique. This gives added possibilities to tailor elements as was described in the case of the Timoshenko beam in section 9.2. However, the original motivation for applying numerical integration was that use of isoparametric mapping caused the element integral to be too complex to be subjected to analytic integration when applied formulated for two- and three-dimensional elements.

All the above will be brought together in the next chapter. There all the steps in formulating an isoparametric element will be demonstrated. The application to a two-dimensional element is sufficient general to expose all steps needed to form any element. Mastering these steps together with the basic equations in the relevant field enables formulation of a wide range of finite elements for linear problems as long as long as the physics does not include additional complexities like nonsymmetric influences. The latter problem is introduced in chapter 13.

The details of the consequences of the isoparametric mapping on the element integrals will shown in the current chapter before going to the element formulation chapters **Error!** **Reference source not found.** and 12. The change of variables will require the determination of the change in scale. This is needed for obtaining derivatives w.r.t the global coordinate system from the derivatives w.r.t the local coordinate system. This information is also needed when evaluating the integrals as the area element $dA=dx dy$ in the integrals is changed.

A two dimensional elements with four nodes is taken as an example. Its geometry is defined in a finite element model with respect to a coordinate system (x,y) . The element integrals are evaluated after a change of variables to a simple coordinate system (s,t) where the element integrals have simple boundaries. This mapping, or variable change, is shown in Figure 11.1.

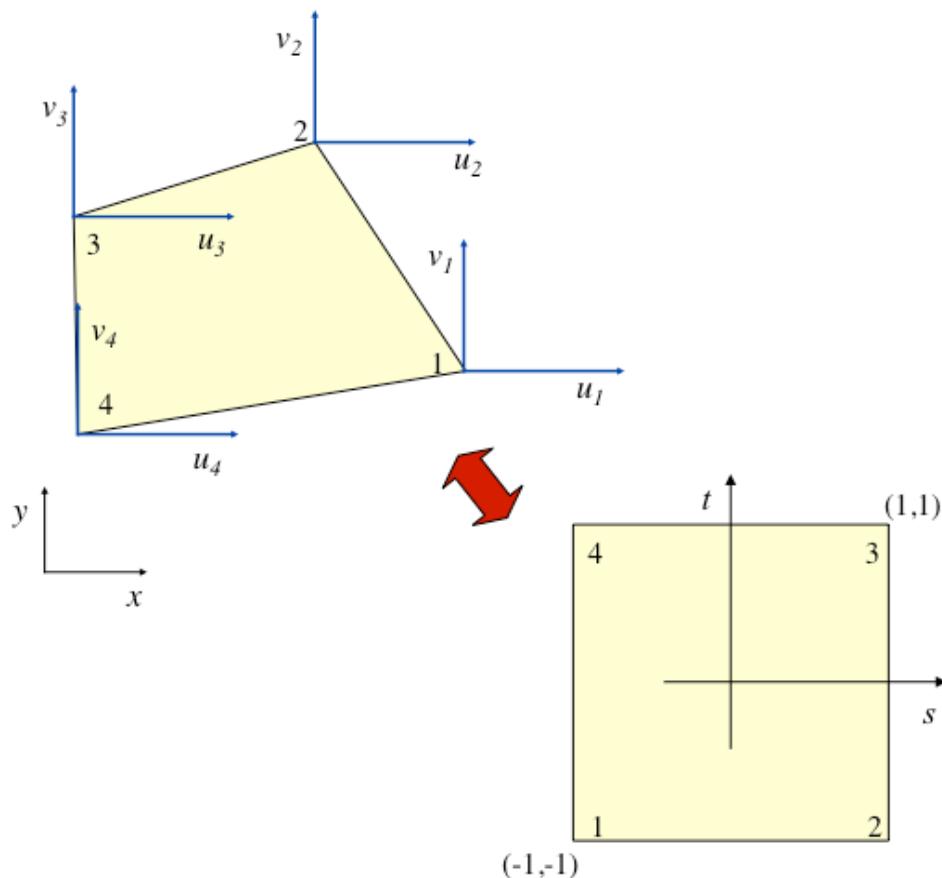


Figure 11.1. Isoparametric mapping of a four node element in two dimensions. The element is a square between $(-1,-1)$ to $(1,1)$ in the local coordinate system.

The same shape functions are used for the interpolating of geometry between the nodes as interpolating the displacements field. The specific shape functions for this element are given in the next chapter but not needed now. The derived formulas will be valid for any kind of element and are easily expanded to the case of three-dimensional integrals. The isoparametric mapping is written as

$$x(s, t) = \sum_{i=1}^{nnode} N(s, t)_i x_i \quad (11.1)$$

$$y(s, t) = \sum_{i=1}^{nnode} N(s, t)_i y_i$$

It can be written in matrix form as

$$\mathbf{x} = \mathbf{Nc} \quad (11.2)$$

The interpolation of the displacement fields is written in the same format, see Eq. (12.1) in next chapter. The derivatives in the equations to be solved are expressed in the global coordinate system. However, the fundamental variables are given with respect to the local coordinates system as the element's shape functions N_i are so defined. The previous, Eq. (11.1), relation between the two coordinate systems is used together with the chain rule. The chain rule states that if we have a function h that is an explicit function of s and s in turn is some kind of function of x . Then it is possible to write

$$\frac{\partial h(s(x, y), t(x, y))}{\partial x} = \frac{\partial h}{\partial s} \frac{\partial s}{\partial x} + \frac{\partial h}{\partial t} \frac{\partial t}{\partial x} \quad (11.3)$$

This can be written as the operator below for the partial derivatives of a function of two coordinates

$$\frac{\partial}{\partial x} = \frac{\partial s}{\partial x} \frac{\partial}{\partial s} + \frac{\partial t}{\partial x} \frac{\partial}{\partial t}$$

$$\frac{\partial}{\partial y} = \frac{\partial s}{\partial y} \frac{\partial}{\partial s} + \frac{\partial t}{\partial y} \frac{\partial}{\partial t}$$

or in matrix form

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial s}{\partial x} & \frac{\partial t}{\partial x} \\ \frac{\partial s}{\partial y} & \frac{\partial t}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} \quad (11.4)$$

However, we can not compute the derivatives $\frac{\partial s}{\partial x}, \frac{\partial s}{\partial y}, \frac{\partial t}{\partial x}$ or $\frac{\partial t}{\partial y}$ as we have only $x(s, t)$ and $y(s, t)$ and not the other way around. Then we set up the derivation in the opposite direction to Eq. (11.4)

$$\begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \quad (11.5)$$

The terms in the Jacobian matrix, \mathbf{J} , can be computed from the isoparametric mapping in Eq. (11.1) as

$$\begin{aligned} J_{11} &= \frac{\partial x}{\partial s} = \frac{\partial}{\partial s} \left(\sum_{i=1}^{nnode} N(s, t)_i x_i \right) = \sum_{i=1}^{nnode} \frac{\partial N_i}{\partial s} x_i, \quad J_{21} = \frac{\partial x}{\partial t} = \sum_{i=1}^{nnode} \frac{\partial N_i}{\partial t} x_i \\ J_{12} &= \frac{\partial y}{\partial s} = \frac{\partial}{\partial s} \left(\sum_{i=1}^{nnode} N(s, t)_i y_i \right) = \sum_{i=1}^{nnode} \frac{\partial N_i}{\partial s} y_i, \quad J_{22} = \frac{\partial y}{\partial t} = \sum_{i=1}^{nnode} \frac{\partial N_i}{\partial t} y_i \end{aligned} \quad (11.6)$$

Then it is possible to invert the expression to give Eq. (9.5).

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = [\mathbf{J}]^{-1} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} \quad (11.7)$$

where

$$|\mathbf{J}| = \det \mathbf{J} = J_{11}J_{22} - J_{12}J_{21} = \frac{\partial x}{\partial s} \frac{\partial y}{\partial t} - \frac{\partial x}{\partial t} \frac{\partial y}{\partial s} \quad (11.8)$$

The above derivations are shown in Figure 11.2 where the information in the derivatives and the determinant of the Jacobian are hopefully clearer. It shows how an infinitesimal area element $dA = ds dt$ becomes a parallelogram. The mapping of a small area element is always a linear mapping.

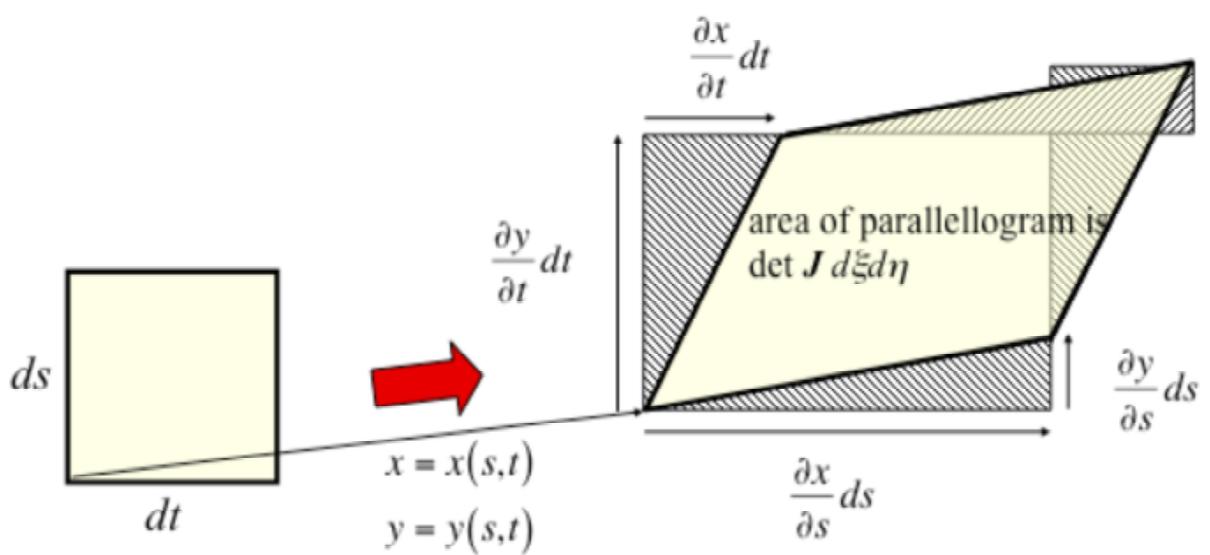


Figure 11.2. Change in scale due to change of variables.

12 Navier's equation in two dimensions

Plane problems

The purpose of this chapter, as stated already in the introduction of the previous chapter, is to bring together all the previous developments into the formulation of a two-dimensional element. An element applicable for solving two-dimensional plane stress³⁹ problems will be derived. The basic continuum mechanics⁴⁰ relations are not given and the reader can consult any standard textbook or book with formulas for solid mechanics or the links below. The formulation steps used are general enough to enable the formulation of elements for several types of linear problems. These general formulas are listed below and thereafter they are detailed for a four node element. The basic relations are also summarised in Figure 12.1

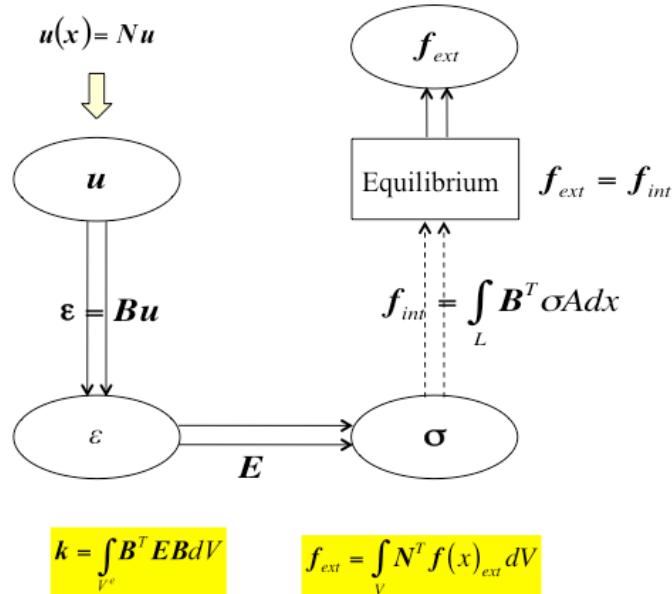


Figure 12.1. The shape functions determine the basic properties of an element. They may be modified by use of underintegration. The diagram shows the formulation steps that follow.

The displacement field within an element is written as

$$\mathbf{u}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{u} \quad (12.1)$$

where the shape functions in the matrix \mathbf{N} fulfils some requirements listed below.

1. The Kronecker delta property. The shape function associated with node J has the property

$$N_J(\mathbf{x}_K) = \begin{cases} 1 & \text{if } J = K \\ 0 & \text{if } J \neq K \end{cases} \quad (12.2)$$

2. The partition of unit property. The summation of the shape functions that overlap at a coordinate x must be unity as given below

$$\sum_J N_J(\mathbf{x}) = 1 \quad (12.3)$$

³⁹ [http://en.wikipedia.org/wiki/Stress_\(physics\)#Plane_stress](http://en.wikipedia.org/wiki/Stress_(physics)#Plane_stress)

⁴⁰ [http://en.wikipedia.org/wiki/Strain_\(materials_science\)](http://en.wikipedia.org/wiki/Strain_(materials_science)), [http://en.wikipedia.org/wiki/Stress_\(physics\)](http://en.wikipedia.org/wiki/Stress_(physics)) , http://en.wikipedia.org/wiki/Strain_tensor#Infinitesimal_strain_tensor, http://en.wikipedia.org/wiki/Hooke%27s_law , [http://en.wikipedia.org/wiki/Deformation_\(mechanics\)](http://en.wikipedia.org/wiki/Deformation_(mechanics))

3. The shape functions must be sufficient continuous. If the fundamental equation to be solved has an $2m$:th derivative, then the WRM formulation after appropriate partial integration⁴¹ will have a derivative of m . Thus it must be possible to take the derivative of the functions at least m times. Furthermore, it must be $m-1$ continuous between elements, C^{m-1} -continuity.

4. The functions must include all lower terms of the family of functions that are used to generate them. This means that for polynomials, all lower order terms must be present before including higher order terms. Furthermore, the different coordinates should be equal present so that the element's properties will not depend on which direction its local coordinate systems has.

The strains needed in the plane stress problem are computed as combinations of derivatives from the displacements fields and this leads to

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u} \quad (12.4)$$

Furthermore, the stresses are related to the strains by Hooke's law in case of linear, elastic problems

$$\boldsymbol{\sigma} = \mathbf{E}\boldsymbol{\epsilon} \quad (12.5)$$

Combining these, see Figure 12.1, leads to a general formula for the stiffness matrix

$$\mathbf{k} = \int_{V^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV \quad (12.6)$$

and the consistent load vector

$$\mathbf{f}_{ext} = \int_V \mathbf{N}^T \mathbf{f}(x)_{ext} dV \quad (12.7)$$

These general relations are shown in detail in the following sections for a four node plane stress element.

12.1 Displacements and geometry of four-node plane element

The local coordinate system of the element is shown in Figure 12.2. This is the basis for formulating the element and where the element integrals, Eq. (12.6) and Eq. (12.7) are solved. The shape functions are

$$\left. \begin{array}{l} N_1 = \frac{1}{4}(1-s)(1-t) \\ N_2 = \frac{1}{4}(1+s)(1-t) \\ N_3 = \frac{1}{4}(1+s)(1+t) \\ N_4 = \frac{1}{4}(1-s)(1+t) \end{array} \right\} \Leftrightarrow N_i = \frac{1}{4}(1+s_is)(1+t_it) \quad (12.8)$$

where the local coordinate (s_i, t_i) is the coordinate of corner i .

⁴¹ This is called geometric compatibility. Some incompatible elements exists but then this incompatibility must disappear as the elements are made smaller in order to guarantee convergence of the solution.

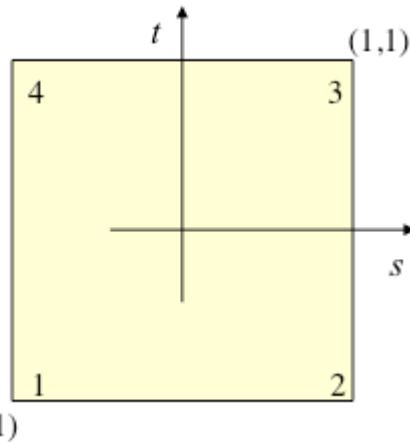


Figure 12.2. Local coordinate system of plane, four node element.

The isoparametric mapping from the local to the global coordinate system, see Figure 11.1, can be written in some variants

$$\mathbf{x} = \begin{bmatrix} x(s,t) \\ y(s,t) \end{bmatrix} = \sum_{i=1}^{nnodes} \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \sum_{i=1}^{nnodes} \mathbf{N}_i \mathbf{c}_i =$$

$$\left[\begin{array}{ccccccccc} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{array} \right] \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \mathbf{Nc} \quad (12.9)$$

The first variant with a sum over *nnodes* sub-matrices is a convenient way to programme the logic for an element with arbitrarily number of nodes. Then follows the specific form for a four node element and finally the general matrix form is given. \mathbf{c} is a vector with nodal coordinates of the element.

Thus the same form (isoparametric) is used to describe the interpolation of the displacements as for the geometry. It is written as

$$\mathbf{u} = \begin{bmatrix} u(s,t) \\ v(s,t) \end{bmatrix} = \sum_{i=1}^{nnodes} \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \sum_{i=1}^{nnodes} \mathbf{N}_i \mathbf{u}_i =$$

$$\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \mathbf{Nu} \quad (12.10)$$

12.2 Strains of four-node plane element

The strain components needed for plane stress are

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x(x,y) \\ \varepsilon_y(x,y) \\ \gamma_{xy}(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \quad (12.11)$$

Insertion of the shape functions, Eq. (12.10), for the four node element gives

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x(s,t) \\ \varepsilon_y(s,t) \\ \gamma_{xy}(s,t) \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} & 0 \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} \quad (12.12)$$

It can be written in the node summation style as

$$\boldsymbol{\varepsilon} = \sum_{i=1}^{nnodes} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \sum_{i=1}^{nnodes} \mathbf{B}_i \mathbf{u}_i = \mathbf{Bu} \quad (12.13)$$

The determination of global derivative of the shape functions are done as shown in chapter 11. The local derivatives of the shape functions are set up first. The Jacobian is determined by Eq. (11.6). Notice that it depends on the coordinates of the element as expected. Thereafter Eq. (11.7) is used to calculate the global derivatives needed for Eq. (12.12). The strains are

explicitly expressed in the local coordinates. If the strain is needed at a coordinate x , then the local coordinate must be solved first from Eq. (12.9) and thereafter the found local coordinate is inserted into Eq. (12.12). Eq. (12.9) can in the general case not be solved analytically but must be solved numerically.

12.3 Constitutive model for plane stress

The relation between strains and stresses, Eq. (12.5), is

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1+\nu} \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (12.14)$$

where E is the modulus of elasticity and ν is Poisson's ratio.

12.4 Flow chart for calculation of four node, plane stress element

All terms for the stiffness matrix and consistent load vectors are now available. The logic for solving the element integrals is given in Box 12.1. It is assumed that the material properties are constant over the element. The \mathbf{B} matrix in the integral for the element stiffness is nearly constant, with exception for one term. The st -term in the shape functions, Eq. (12.8), gives one linear term. Thus most of the terms in the integral for the stiffness matrix, Eq. (12.6), are constant or linear. However, the highest term is quadratic. This requires $nint = 2*2$ integration points according to Eq. (7.2).

Box 12.1. Procedure for the computation of element stiffness matrix.

Set up constitutive matrix \mathbf{E} , Eq. (12.14).

For $igaus$ from 1 to $nint$ integration points

Look up the location and weight of current point integration point s_{igaus} , t_{igaus} and w_{igaus} ⁴²

Evaluate the shape functions $N_{inode}(s_{igaus}, t_{igaus})$ and their local derivatives

$$\left. \frac{\partial N_{inode}}{\partial s} \right|_{igaus}, \left. \frac{\partial N_{inode}}{\partial t} \right|_{igaus} \text{ with } inode \text{ from 1 to } nnodes$$

Compute the Jacobian matrix, Eq. (11.6), and its determinant, Eq. (11.8), at the integration point

$$J_{11} = \sum_{inode=1}^{nnodes} \left(\left. \frac{\partial N_{inode}}{\partial s} \right|_{igaus} \cdot x_{inode} \right), J_{21} = \sum_{inode=1}^{nnodes} \left(\left. \frac{\partial N_{inode}}{\partial t} \right|_{igaus} \cdot x_{inode} \right)$$

$$J_{12} = \sum_{inode=1}^{nnodes} \left(\left. \frac{\partial N_{inode}}{\partial s} \right|_{igaus} \cdot y_{inode} \right), J_{22} = \sum_{i=1}^{nnodes} \left(\left. \frac{\partial N_{inode}}{\partial t} \right|_{igaus} \cdot y_{inode} \right)$$

$$\det \mathbf{J}_{igaus} = J_{11}J_{22} - J_{12}J_{21}$$

Compute the global derivatives of the shape functions according to Eq. (11.4)

⁴² This is the weight produced by multiplication of two appropriate weights according to Eq. (7.5).

$$\begin{bmatrix} \frac{\partial N_{inode}}{\partial x} \\ \frac{\partial N_{inode}}{\partial y} \end{bmatrix}_{igaus} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_{inode}}{\partial s} \\ \frac{\partial N_{inode}}{\partial t} \end{bmatrix}_{igaus}$$

Place the global derivatives at appropriate positions in the **B**-matrix. Evaluate the contribution to the element integrals from the current integration points

$\mathbf{B}_{igaus}^T \mathbf{E} \mathbf{B}_{igaus} \det \mathbf{J}_{igaus} w_{igaus}$ and sum this into k.

Next integration point

13 Convective heat transfer – non-symmetric physics

The previous equations are quite straightforward to solve using the finite element method. There are other equations that, even in the linear case, need special precautions in order to be tractable. One example discussed below is the inclusion of a convective term in the heat conduction equation. This equation describes the combined conduction and convection of heat in a medium with a given flow where the material moves with respect to the coordinate system. The equation is also obtained when introducing a moving coordinate system in the heat conduction equation. Then the coordinate system is moving with respect to the material.

The one-dimensional heat conduction equation⁴³ in absence of heat sinks is written as

$$\rho c \dot{T} - \lambda \frac{d^2 T}{dx^2} = 0 \quad (13.1)$$

where χ is the coordinate, ρ is the density, c is the heat capacity and λ is the heat conductivity. Then we switch to a moving coordinate system, x , by

$$x = \chi + vt \quad (13.2)$$

where v is the velocity. This transforms⁴⁴ Eq. (13.1) to

$$\rho c \left(\dot{T} + v \frac{dT}{dx} \right) - \lambda \frac{d^2 T}{dx^2} = 0 \quad (13.3)$$

Furthermore, the solution is assumed to be steady-state w.r.t. this moving coordinate system leading to

$$\rho c v \frac{dT}{dx} - \lambda \frac{d^2 T}{dx^2} = 0 \quad (13.4)$$

There is an analytic solution of this problem with the boundary conditions

$$T(0) = T_{left} \quad (13.5)$$

$$T(L) = T_{right} \quad (13.6)$$

The solution can be expressed by introducing the Peclet number. It measures the relative strength of convection and conduction of heat. It requires a reference length and we use the length of an element l^e for this. The number becomes

$$Pe = \frac{\rho c v l^e}{2\lambda} \quad (13.7)$$

The analytic solution can now be written as

$$T(x) = T_{left} + (T_{right} - T_{left}) \frac{\left(e^{\frac{2Pe x}{l^e}} - 1 \right)}{\left(e^{\frac{2Pe L}{l^e}} - 1 \right)} \quad (13.8)$$

The finite element solution leads to a global system of equations

$$\mathbf{K}\mathbf{T} = \mathbf{0} \quad (13.9)$$

⁴³ http://en.wikipedia.org/wiki/Heat_conduction

⁴⁴ http://en.wikipedia.org/wiki/Total_derivative

The standard contribution to \mathbf{K} from the conductivity matrix using a two-node linear element is, see section 15.6,

$$\mathbf{k}_{cond} = \int_{v(e)} \mathbf{B}^T \lambda \mathbf{B} dv = \frac{\lambda}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (13.10)$$

The additional contribution due to the convection part can be derived by comparing the convective term with the u -term in Eq. (6.1) or by comparing it with the q -term in Eq. (9.2). Using the latter leads to an expression for consistent nodal loads shown in Eq. (9.14). Thus q is replaced by $\rho cv \frac{dT}{dx}$ in this equation. This gives

$$\int_{-1}^1 \mathbf{N}^T \rho cv \frac{dT}{dx} J ds \quad (13.11)$$

The gradient term is

$$\mathbf{BT}^e = \frac{1}{l^e} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}^e \quad (13.12)$$

Combining the last two equations gives

$$\int_{-1}^1 \rho cv \mathbf{N}^T \mathbf{B} J ds \mathbf{T}^e = \mathbf{k}_{conv} \mathbf{T}^e = \frac{\rho cv}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \mathbf{T}^e \quad (13.13)$$

Thus it will also contribute to the \mathbf{K} -matrix. Summing the two matrices, Eq. (13.12) and Eq. (13.13), and using Eq. (13.7) gives the combined element matrix

$$\mathbf{k} = \frac{\lambda}{l^e} \left[\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + Pe \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \right] = \frac{\lambda}{l^e} \begin{bmatrix} 1-Pe & -1+Pe \\ -1-Pe & 1+Pe \end{bmatrix} \quad (13.14)$$

to be assembled into Eq (13.9). Note that the matrix is not symmetric. This formulation becomes unstable when $Pe=1$ as can be seen in the example below. This numerical problem can be reduced by enhancing the conductivity so that the Peclet number is modified to $_{mod}Pe$ according to the equation below

$$_{mod}Pe = \tanh(Pe) \quad (13.15)$$

The classical formulation is compared with the stabilised formulation and theoretical solutions for $Pe=1$ in Figure 13.1 and $Pe=3$ in Figure 13.2.

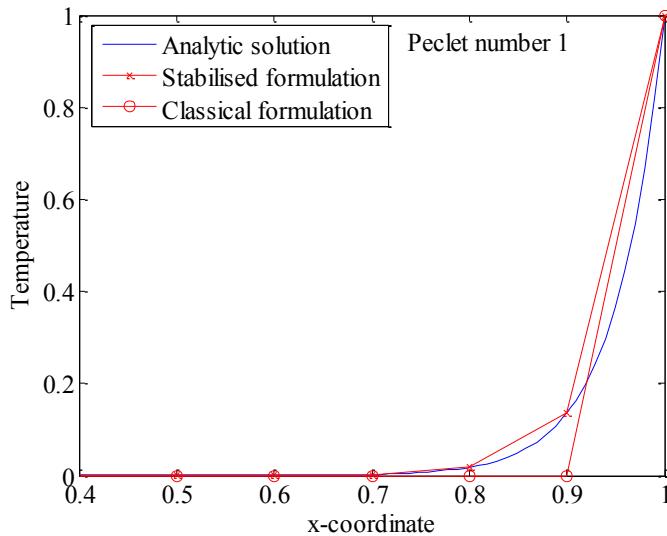


Figure 13.1. Temperature in fluid flow with given inlet and outlet temperatures for $Pe=1$.

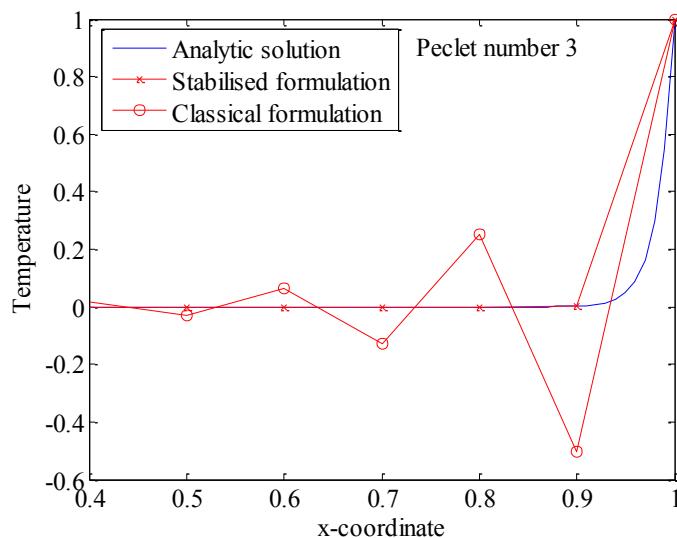


Figure 13.2. Temperature in fluid flow with given inlet and outlet temperatures for $Pe=3$.

A comparison of the methods for different Peclet numbers and number of elements is shown in the table below.

Table 13.1. L2-error and condition number of matrix for conduction-convection heat transfer problem using $L=\rho=\lambda=c=1$ and varying velocity giving varying Peclet numbers.

Number of elements	Peclet number	Standard method		Stabilised method	
		Error	Condition number	Error	Condition number
10	0.5	0.004	27.6	7.6e-18	28.8
10	1	0.034	18	2.7e-18	20.5
10	2	0.034	18.5	1.3e-18	18.3
10	4	0.07	18.9	4.4e-19	18.
10	8	0.13	20.6	2.1e-18	18.
10	16	0.26	27.	6.9e-19	18.

The heat transfer in a material moving with the velocity v in positive direction and a point source at the coordinate x_c is

$$\rho c v \frac{dT}{dx} - \lambda \frac{d^2T}{dx^2} = P \delta(x - x_c) \quad (13.16)$$

where v is now interpreted as the material flow relative to the heat source. Thus the heat source is moving towards the left in the figure below. The boundary conditions and the finite element formulations are taken as the same as in the previous example. It is assumed that the point source is applied on a node. Then there will be one contribution to the load vector and Eq. (13.9) becomes

$$\mathbf{K}_{th} \mathbf{T} = \dot{\mathbf{Q}}_{ext} \quad (13.17)$$

The enhancement of the heat conductivity that gives $_{mod} Pe$ in Eq. (13.15) is

$$_{mod} \lambda = \lambda(1 + \alpha Pe) \quad (13.18)$$

where

$$\alpha = \coth(Pe) - \frac{1}{Pe} \quad (13.19)$$

The value $P/(1 + \alpha Pe)$ is placed at the appropriate position in $\dot{\mathbf{Q}}_{ext}$ corresponding to the node at x_c . The classical formulation is compared with the stabilised formulation and theoretical solutions for $Pe=1$ in Figure 13.3 and $Pe=3$ in Figure 13.4.

The improved formulation using an enhanced convection is a prelude to Stream Upwind Petrov-Galerkin (SUPG) formulations. This it is can be derived using a Petrov-Galerkin approach.

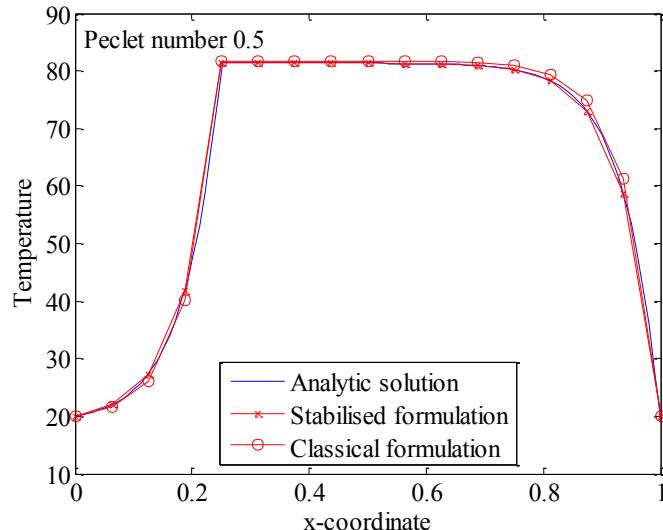


Figure 13.3. Temperature due to a point heat source moving to the left, $Pe=0.5$.

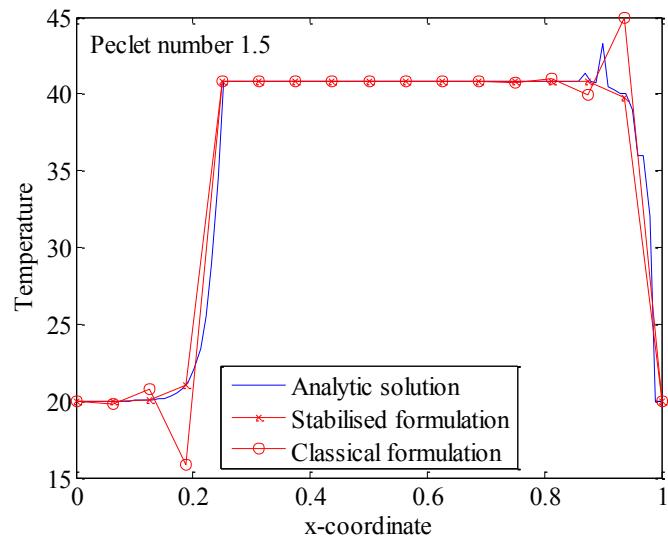


Figure 13.4. Temperature due to a point heat source moving to the left, $Pe=1.5$. Wiggles at end of analytic solution due to inaccuracy in its evaluation.

14 References

1. Adams, V. and A. Askenazi, *Building Better Products with Finite Element Analysis*. 1999: OnWord Press.
2. Hutton, D., *Fundamentals of Finite Element Analysis*. 2004: McGraw-Hill.
3. Belytschko, T., W.K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. 2000, Chichester, England: John Wiley & Sons. 650.
4. Fletcher, C., *Computational Galerkin Methods*. Springer Series in Computational Physics, ed. H. Cabannes, et al. 1984, New-York: Springer-Verlag. 320.

15 Appendix.

15.1 Tensor and vector notations in mechanical problems

The strains and stresses in mechanical problems are second order tensors⁴⁵ that have special properties for how they change when the coordinate system is changed. The major difference is how shear strain is defined. They can be stored in matrices or as vectors. We write them out for the two-dimensional case below. The strains are obtained from displacements as indicated in Figure 3.4 by

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u}^T + \mathbf{u} \nabla^T) \quad (15.1)$$

when using tensor notation. This gives

$$\begin{bmatrix} \boldsymbol{\varepsilon}_x & \boldsymbol{\varepsilon}_{xy} \\ \boldsymbol{\varepsilon}_{xy} & \boldsymbol{\varepsilon}_y \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} u & v \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \right) = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \frac{\partial v}{\partial y} \end{bmatrix} \quad (15.2)$$

The use of vector notation is often, but not always, combined with engineering definition of shear and becomes

$$\begin{bmatrix} \boldsymbol{\varepsilon}_x \\ \boldsymbol{\varepsilon}_y \\ \gamma_{xy} \end{bmatrix} = \nabla \mathbf{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \end{bmatrix}. \quad (15.3)$$

The difference between the shear definitions is

$$\gamma_{xy} = 2\boldsymbol{\varepsilon}_{xy}. \quad (15.4)$$

The equilibrium equations in Figure 3.4 for the tensor variant are written as

$$\begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \sigma_x & \sigma_{xy} \\ \sigma_{xy} & \sigma_y \end{bmatrix} + \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \rho \frac{\partial^2}{\partial t^2} \begin{bmatrix} u \\ v \end{bmatrix} \quad (15.5)$$

and for the vector variant

$$\begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} + \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \rho \frac{\partial^2}{\partial t^2} \begin{bmatrix} u \\ v \end{bmatrix}. \quad (15.6)$$

⁴⁵ <http://en.wikipedia.org/wiki/Tensor>

Notice that this affects also the constitutive relation. The Hooke's law cannot be written in a matrix format when using tensor notation as it is a fourth order tensor. It is only written in matrix form when strains and stresses are stored in vector format. This format is also easier to base an implementation of the finite element formulation on. The relation between shear strain and stresses differs. The tensor definition of shear has

$$\sigma_{xy} = 2G\epsilon_{xy}. \quad (15.7)$$

G is the shear modulus. The engineering definition is

$$\sigma_{xy} = G\gamma_{xy}. \quad (15.8)$$

15.2 Galerkin and Ritz methods

The relation between the Galerkin method and Ritz methods can be described as follows. One can note that the Galerkin method is a variational method whereas as Ritz method is a minimisation method. Assume that u is the solution to the differential equation

$$A(u) = f_{ext} \quad (15.9)$$

where A is a positive definite operator. This property means that

$$\int_{\Omega} u A(u) dv > 0 \text{ for all } u \neq 0 \quad (15.10)$$

Then the solution to Eq. (15.9) can be shown to be equivalent to finding the minimum of the functional

$$\Pi(u) = \frac{1}{2} \int_{\Omega} u A(u) dv - \int_{\Omega} u f_{ext} dv \quad (15.11)$$

An approximate solution like in Eq. (4.7) is used but now it only fulfils the essential boundary conditions as in Eq. (4.17). The relation above is called the principle of minimum total potential energy⁴⁶ in mechanics. The field u must be admissible. This is fulfilled if the gradient is computed correctly from this field and if it fulfils essential boundary conditions.

Assuming that the latter can be fulfilled by fixing appropriate coefficients a_i leads to

$$\Pi(\hat{u}) = \frac{1}{2} \int_{\Omega} \sum_{k=1}^N a_k(t) \varphi_k(x) \sum_{i=1}^N a_i(t) A(\varphi_i(x)) dv - \int_{\Omega} \sum_{k=1}^N a_k(t) \varphi_k(x) f_{ext} dv \quad (15.12)$$

This can be written in matrix form as

$$\Pi(\hat{u}) = \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{a} - \mathbf{a}^T \mathbf{F}_{ext} \quad (15.13)$$

where the coefficients of the matrix and vector are

$$K_{ki} = \int_{\Omega} \varphi_k(x) A(\varphi_i(x)) dv \quad (15.14)$$

and

$$F_{ext,k} = \int_{\Omega} \varphi_k(x) f_{ext} dv \quad (15.15)$$

The best choice of parameters is the set that minimise this functional. A condition for this is

$$\frac{\partial \Pi(\hat{u})}{\partial a_k} = 0 \quad \text{for } k = 1, \dots, N \quad (15.16)$$

This leads to

⁴⁶ http://en.wikipedia.org/wiki/Minimum_total_potential_energy_principle

$$\mathbf{K}\mathbf{a} = \mathbf{F}_{ext} \quad (15.17)$$

This is the same as for the Galerkin method as can be seen in section 4.8.7. The connection between Galerkin method and Ritz method is that convergence properties for the latter method are also valid for Galerkin methods. They state that increasing N makes the functional Π go towards the true minimum. Thus an approximate solution will not reach down to the true minimum but we will have convergence from above in terms of the norm. This norm can be interpreted as an energy norm in mechanical problems.

15.3 Condensation of a system of equations.

A constrain between unknown coefficients can be written as

$$\mathbf{C}\mathbf{U} - \mathbf{Q} = \mathbf{0} \Leftrightarrow \begin{bmatrix} \mathbf{C}_r & \mathbf{C}_c \end{bmatrix} \begin{bmatrix} \mathbf{U}_r \\ \mathbf{U}_c \end{bmatrix} - \mathbf{Q} = \mathbf{0}, \quad (15.18)$$

where the matrix \mathbf{C} and vector \mathbf{Q} contains constants. The coefficients in \mathbf{U}_c will be condensed out and those in \mathbf{U}_r will be retained. Furthermore, we can rewrite it as

$$\mathbf{C}_r \mathbf{U}_r + \mathbf{C}_c \mathbf{U}_c = \mathbf{Q} \Rightarrow \mathbf{U}_c = \mathbf{C}_c^{-1} [\mathbf{Q} - \mathbf{C}_r \mathbf{U}_r]. \quad (15.19)$$

Then the vector with unknown coefficients can be written as

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_r \\ \mathbf{U}_c \end{bmatrix} = \begin{bmatrix} \mathbf{U}_r \\ \mathbf{C}_c^{-1} [\mathbf{Q} - \mathbf{C}_r \mathbf{U}_r] \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}_c^{-1} \mathbf{C}_r \end{bmatrix} \mathbf{U}_r + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}_c^{-1} \mathbf{Q} \end{bmatrix}, \quad (15.20)$$

or shortly

$$\mathbf{U} = \mathbf{T}\mathbf{U}_r + \mathbf{Q}_0. \quad (15.21)$$

Assume that we want to solve

$$\mathbf{K}\mathbf{U} = \mathbf{F}_{ext}. \quad (15.22)$$

Then the introduction of the constraint gives

$$\mathbf{K}\mathbf{T}\mathbf{U}_r = \mathbf{F}_{ext} - \mathbf{K}\mathbf{Q}_0. \quad (15.23)$$

Furthermore, we can shrink (condense) the system by pre-multiplying left and right hand sides with \mathbf{T}^T giving

$$\mathbf{T}^T \mathbf{K} \mathbf{T} \mathbf{U}_r = \mathbf{T}^T (\mathbf{F}_{ext} - \mathbf{K} \mathbf{Q}_0) \quad (15.24)$$

or shorter

$$\mathbf{K}_r \mathbf{U}_r = \mathbf{F}_{mod}. \quad (15.25)$$

The reduced matrix will be symmetric if the original matrix \mathbf{K} is symmetric. It is possible to get the remaining coefficients after solving for \mathbf{U}_r by

$$\mathbf{U}_c = \mathbf{C}_c^{-1} [\mathbf{Q} - \mathbf{C}_r \mathbf{U}_r]. \quad (15.26)$$

15.4 Bernoulli beam

The classical beam theory assumes that a line orthogonal to the centre line of the beam remains straight and orthogonal as shown in Figure 15.1. Then the axial displacement due to a deflection of the centre line is

$$u = -z(-\theta) = z\theta = -z \frac{dw}{dx} \quad (15.27)$$

This gives a linear variation of the axial strain as

$$\varepsilon = \frac{du}{dx} = z \frac{d\theta}{dx} = -z \frac{d^2 w}{dx^2} = z\kappa \quad (15.28)$$

All other strains are zero due to the assumed deformation in Eq. ((15.27)). The bending moment corresponding to the linear axial stress distribution, see Figure 15.2, becomes

$$M = \int_A \sigma z dA = E \int_A \varepsilon z dA = -E \frac{d^2 w}{dx^2} \int_A z^2 dA \quad (15.29)$$

The last integral is the definition of area moment of inertia giving

$$M = -EI \frac{d^2 w}{dx^2} = EI\kappa \quad (15.30)$$

The curvature has been introduced above as

$$\kappa = -\frac{d^2 w}{dx^2} \quad (15.31)$$

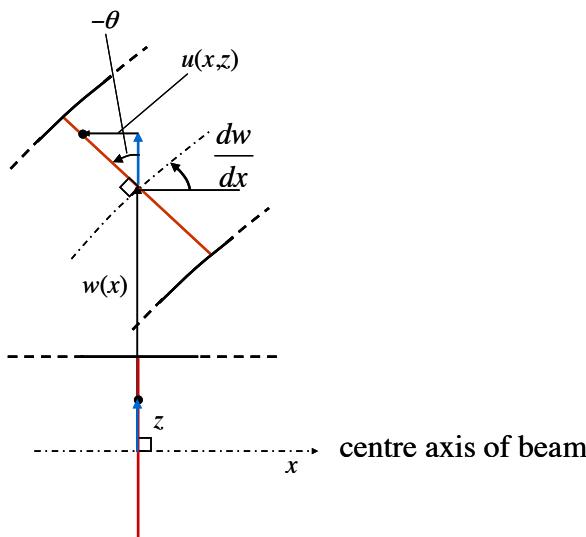


Figure 15.1. Assumed deformation of beam cross-section according to Bernoulli beam theory.

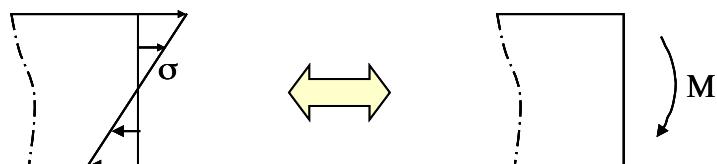


Figure 15.2. Axial stress distribution and equivalent bending moment.

Moment equilibrium for an infinitesimal element, Figure 15.3, gives

$$M - \left(M + \frac{dM}{dx} dx \right) + T \frac{dx}{2} + \left(T + \frac{dT}{dx} dx \right) \frac{dx}{2} = 0$$

leading to

$$\frac{dM}{dx} = T \quad (15.32)$$

Transverse force equilibrium gives

$$-T + \left(T + \frac{dT}{dx} dx \right) + qdx = 0$$

leading to

$$\frac{dT}{dx} = -q \quad (15.33)$$

Combining Eq. (15.32) and Eq. (15.33) gives

$$\frac{d^2M}{dx^2} = -q \quad (15.34)$$

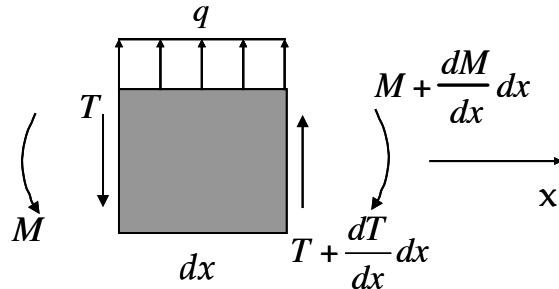


Figure 15.3. Equilibrium of infinitesimal piece of beam.

Insertion of Eq.(15.30) gives

$$\frac{d^4w}{dx^4} = \frac{q}{EI} \quad (15.35)$$

This is the equilibrium equation for a beam expressed in the displacement field $w(x)$. The basic relations are summarised in Figure 15.4. Notice that there are transverse forces playing a role in the equilibrium equations. They are related to shear stresses on the cross-section. BUT the assumption about the deformation of the cross-section states that there is no shear straining. This is implied in the statement that the line orthogonal to the centre line remains orthogonal.

There exist corresponding formulations for plates and shells. Assumptions about the deformation behaviour with respect to a reference surface reduce the dimension of the problem but raise the order of the differential equation.

The boundary conditions for the beam are of the following types;

- Essential boundary conditions at one of the ends are prescribed deflection w or rotation $\frac{dw}{dx}$.
- Natural boundary conditions are prescribed end moment or transverse force. They correspond to $\frac{d^2w}{dx^2}$ or $\frac{d^3w}{dx^3}$ according to Eq.s (15.30) and (15.32).

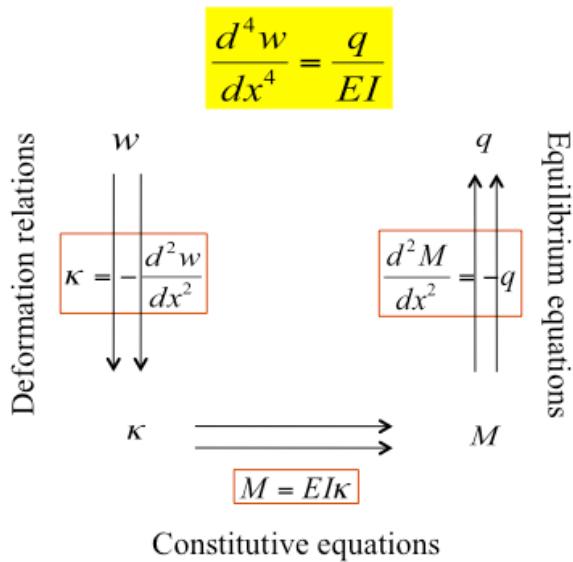


Figure 15.4. Tonti diagram for beam relations

The total potential energy of the beam has a contribution from axial stresses and external loading. Assuming that we have only distributed load q gives

$$\Pi = \iint_{L A} \frac{1}{2} \sigma \epsilon dA dx - \int_L q w dx = \iint_{L A} \frac{1}{2} E \epsilon^2 dA dx - \int_L q w dx$$

Hooke's law, $\sigma = E \epsilon$, was used in the last step above. There is no need to include shear stresses as the corresponding strains are zero and therefore do not contribute to the stored elastic energy in the first integral. Insertion of Eq. (12.4) gives

$$\Pi = \iint_{L A} \frac{1}{2} E \left(-z \frac{d^2 w}{dx^2} \right)^2 dA dx - \int_L q w dx$$

Some quantities does only vary with the x-coordinate

$$\Pi = \int_L \frac{1}{2} E \left(\frac{d^2 w}{dx^2} \right)^2 \int_A z^2 dA dx - \int_L q w dx$$

Use of the definition of area moment of inertia gives finally

$$\Pi = \int_L \frac{1}{2} EI \left(\frac{d^2 w}{dx^2} \right)^2 dx - \int_L q w dx \quad (15.36)$$

A cantilever beam problem with a point load at the end as shown in Figure 15.5 is defined by Eq. (15.35) and the appropriate boundary conditions;

no displacement near wall

$$w(0) = 0 \quad (15.37)$$

and no slope near wall

$$\left. \frac{dw}{dx} \right|_{x=0} = 0 \quad (15.38)$$

No moment at right edge

$$\left. \frac{d^2 w}{dx^2} \right|_{x=L} = 0 \quad (15.39)$$

Given transverse force at right edge

$$T = -EI \frac{d^3 w}{dx^3} \Big|_{x=L} = P \quad (15.40)$$

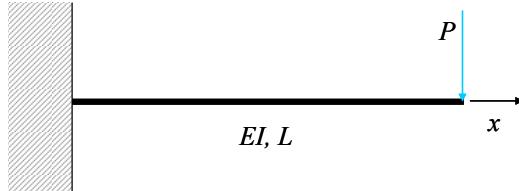


Figure 15.5. Cantilever beam with point load.

Integration of Eq. (15.35) four times with $q=0$ gives

$$w = A + Bx + Cx^2 + Dx^3 \quad (15.41)$$

Thus the exact solution for a beam loaded only at its edges is no higher than a cubic polynomial. This observation can be related to the beam element formulation in chapter Figure 9.1. The two first boundary condition at $x=0$ gives $A=B=0$.

The boundary condition Eq. (15.40) gives

$$D = \frac{P}{6EI} \quad (15.42)$$

and the use of Eq. (15.39) leads to

$$2C + 6DL = 2C + \frac{PL}{EI} = 0 \rightarrow C = \frac{PL}{2EI} \quad (15.43)$$

The solution is thus

$$w(x) = -\frac{PL}{2EI}x^2 + \frac{P}{6EI}x^3 = \frac{PL^3}{EI} \left(\frac{1}{6} \left(\frac{x}{L} \right)^3 - \frac{1}{2} \left(\frac{x}{L} \right)^2 \right) \quad (15.44)$$

The maximum displacement (downwards) is

$$w_{BBT} = -\frac{PL^3}{3EI} \quad (15.45)$$

15.5 Timoshenko beam

This beam theory is an extension of the previously described Bernoulli beam theory. It assumes that a line orthogonal to the centre line of the beam remains straight but not orthogonal as shown in Figure 15.1 but now as in Figure 15.6. Thus we allow for a constant shear strain over the thickness of the beam. This is a step forward. Previously it was noted that the transverse force corresponds to shear stresses. Shear stresses are obtained from shear strains by the shear modulus G as

$$\tau = G\gamma = \frac{E}{2(1+\nu)}\gamma \quad (15.46)$$

However, analysis not included above shows that the shear stresses are largest in the interior of the beam and zero at the top and bottom surfaces. Thus the shear strains are also varying with z . The Timoshenko beam theory includes shear, γ , but assumes it to be constant over the height of the beam. Then the axial displacement due to a deflection of the centre line is

$$u = -z(-\theta) = z\theta = -z\left(\frac{dw}{dx} - \gamma\right) \quad (15.47)$$

This gives a linear variation of the axial strain as

$$\varepsilon = \frac{du}{dx} = z \frac{d\theta}{dx} \quad (15.48)$$

The axial strain cannot be related to the second derivative w anymore. Accordingly the formula for the cross-sectional moment, see Figure 15.2, becomes somewhat different than for the Bernoulli beam. It becomes

$$M = \int_A \sigma z dA = E \int_A \varepsilon z dA = E \left(\frac{d\theta}{dx} \right) \int_A z^2 dA = EI \frac{d\theta}{dx} \quad (15.49)$$

Now the relation between shear stresses and transverse force, see Figure 15.7, is needed.

$$T = \int_A \tau dA = \int_A G \gamma_{xy} dA \approx \frac{GA\gamma}{\alpha} = GA_s \gamma \quad (15.50)$$

The parameter α was introduced for enabling the simplification of constant shear over the cross-section. It gives an effective shear area

$$A_s = \frac{A}{\alpha} \quad (15.51)$$

The value depends on the shape of the cross-section of the beam and is tabulated.

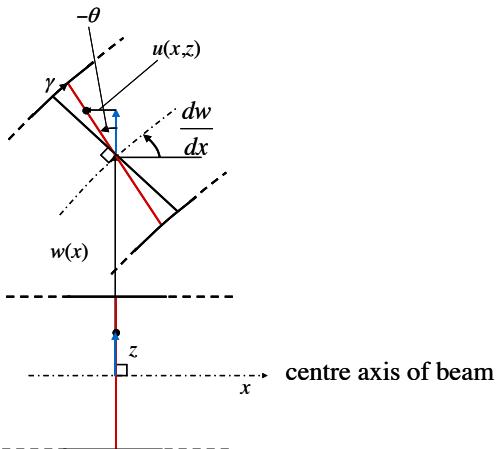


Figure 15.6. Assumed deformation of beam cross-section according to Timoshenko beam theory.

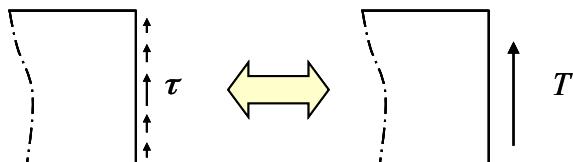


Figure 15.7. Transverse force and shear stresses.

The equilibrium relations for an infinitesimal element, Figure 15.3, are the same as for the Bernoulli beam giving

$$\frac{dM}{dx} = T \quad (15.52)$$

and

$$\frac{dT}{dx} = -q \quad (15.53)$$

Combining them gives

$$\frac{d^2M}{dx^2} = -q \quad (15.54)$$

There exist corresponding formulations for plates and shells and then the theories are called thick plates/thick shells. Sometimes the plate theory is called Mindlin plate theory. The basic relations for the Timoshenko beam are shown in Figure 15.8.

We will not formulate the equilibrium equation corresponding to Eq. (15.35) but proceed directly to the expression for the potential energy. The elastic energy is now a sum of energy due to shear and normal deformations.

$$\Pi_e = \int_V \left(\frac{1}{2} \sigma \epsilon + \frac{1}{2} \tau \gamma \right) dA dx \quad (15.55)$$

Insertion of Eq. (15.48) and Eq. (15.49) in the integral above and integrating over the cross-section gives

$$\Pi_e = \int_L \left(\frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 + \frac{1}{2} \frac{GA}{\alpha} \gamma^2 \right) dx \quad (15.56)$$

The total potential energy of the beam has a contribution from axial stresses and external loading. Assuming that we have only distributed load q gives

$$\Pi = \int_L \frac{1}{2} \begin{bmatrix} \frac{d\theta}{dx} & \gamma \end{bmatrix} \begin{bmatrix} EI & 0 \\ 0 & \frac{GA}{\alpha} \end{bmatrix} \begin{bmatrix} \frac{d\theta}{dx} \\ \gamma \end{bmatrix} dx - \int_L q w dx \quad (15.57)$$

The basic relations are summarised below. The vectors \mathbf{f}_{ext} and \mathbf{u} will be introduced when formulating a finite element in chapter 9.2.

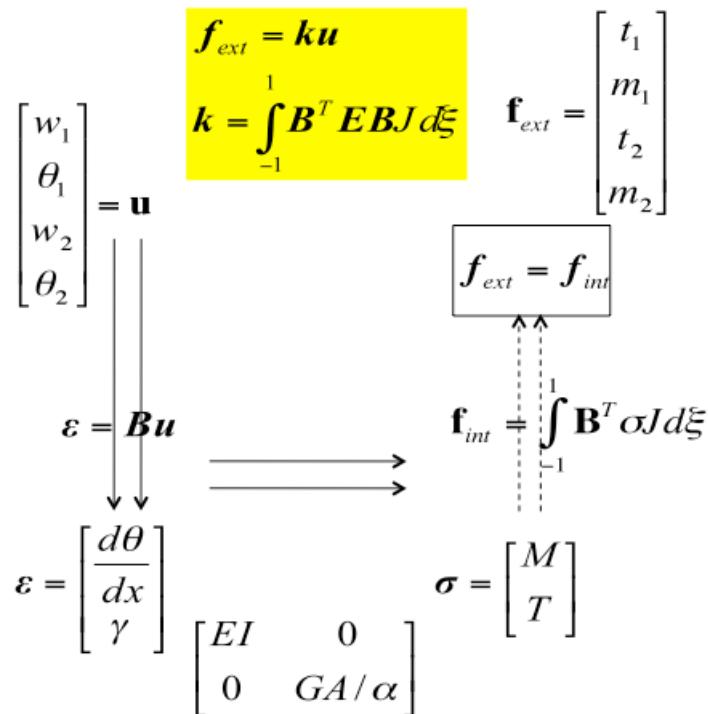


Figure 15.8. Tonti diagram for Timoshenko beam theory.

The solution according to Timoshenko beam theory of the deflection of the beam in Figure 15.5 with a square cross-section is

$$w(x) = \frac{PL^3}{EI} \frac{1}{2} \left(\frac{x}{L} \right)^2 \left[\frac{1}{3} \left(\frac{x}{L} \right) - 1 \right] - \frac{P\alpha}{kGA} x$$

$$G = \frac{E}{2(1+\nu)} \text{ and for } \alpha = \frac{12+11\nu}{10(1+\nu)}$$

This gives the maximum deflection

$$\begin{aligned} w_{TBT} &= -\frac{PL^3}{3EI} - \frac{PL}{GA} \frac{(12+11\nu)}{10(1+\nu)} = -\frac{12PL^3}{3EBH^3} - \frac{PL}{GBH} \frac{(12+11\nu)}{10(1+\nu)} \\ w_{LBB} &= -\frac{PL^3}{3EI^3} \left[1 + \left(\frac{H}{L} \right)^2 \frac{(12+11\nu)}{20} \right] \end{aligned} \quad (15.58)$$

15.6 One-dimensional element for heat conduction

The one-dimensional heat conduction equation⁴³ is written as

$$L(T) = \lambda \frac{d^2T}{dx^2} + \dot{Q} - \rho c \dot{T} = 0 \quad (15.59)$$

where ρ is the density, c is the heat capacity and λ is the heat conductivity. A Tonni diagram for the basic relations is shown in Figure 15.9.

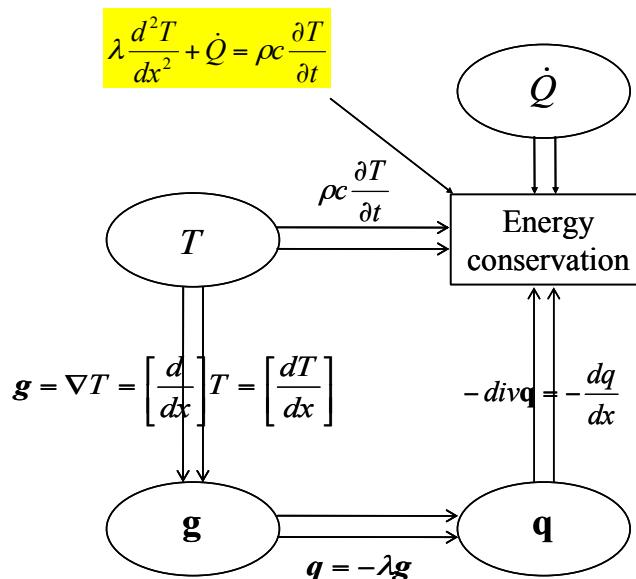


Figure 15.9. Tonni diagram for heat conduction equation.

The finite element equations corresponding to Figure 15.9 are shown in Figure 15.11. They can be derived using the Weighted Residual Method following the same steps as in chapter 6.

The temperature in an element is interpolated by

$$T(s) = \mathbf{NT}^e = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}^e \quad (15.60)$$

where the finite element shape functions for a two node element, see Figure 15.10, are the same as in section 6.4

$$N_1 = \frac{1}{2}(1-s) \quad N_2 = \frac{1}{2}(1+s) \quad (15.61)$$

The element conductivity matrix is

$$\mathbf{k} = \int \mathbf{B}^T \lambda \mathbf{B} J A d s \mathbf{T}^e = \int_{-1}^1 \lambda \frac{1}{l^e} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \frac{1}{l^e} \begin{bmatrix} -1 & 1 \end{bmatrix} A \frac{l^e}{2} ds \quad (15.62)$$

This gives

$$\mathbf{k} = \frac{\lambda A}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (15.63)$$

The element heat capacity matrix is

$$\mathbf{c} = \int_{-1}^1 \rho c \mathbf{N}^T \mathbf{N} J A d s = \frac{\rho c A l^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (15.64)$$

The use of a lumped heat capacity matrix can be advantageous sometimes. It is

$$\mathbf{c} = \frac{\rho c A l^e}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (15.65)$$

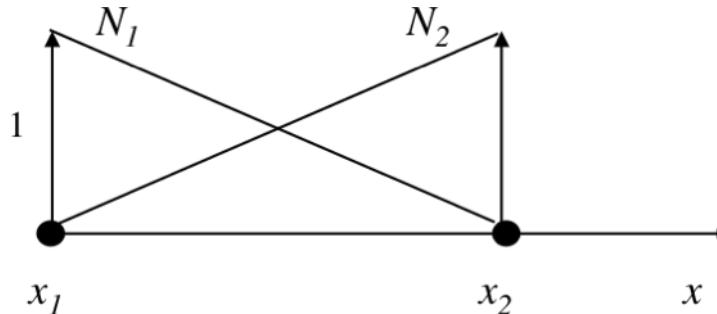


Figure 15.10. Two node element and its linear shape functions N_1 and N_2 .

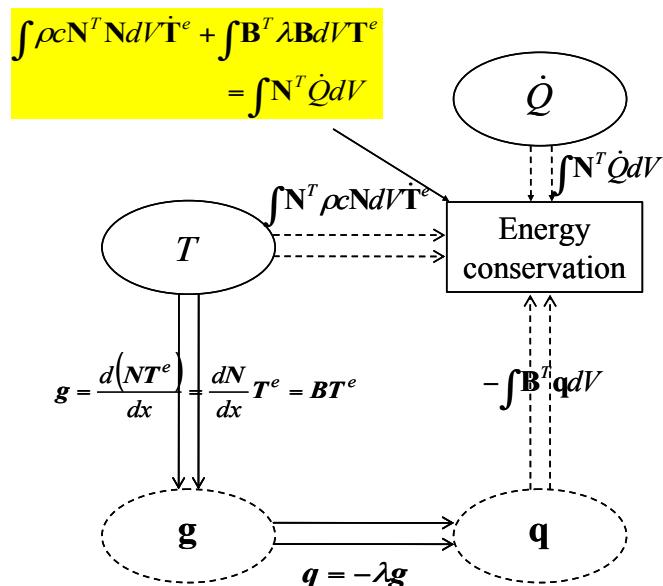


Figure 15.11. Tonti diagram for finite element relations for heat conduction.