From Weighted Residual Methods to Finite Element Methods

Lars-Erik Lindgren

2009

TABLE OF CONTENT

1 Introduction

The finite element method is a general method for solving partial differential equations of different types. It has become a standard method in industry for analysing thermo-mechanical problems of varying types. It has to a large extent replaced experiments and testing for quick evaluation of different design options.

This text is supplementary material in an undergraduate course about modelling in multiphysics. The aim is to show how a finite element formulation of a given mathematical problem can be done. Naturally, the focus is on simple, linear equations but some discussions of more complex equations with convective terms are also included.

2 Some 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 can not 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 use to compute the results at one instant of time.

3 Short finite element course

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 see [1] and see [2] for its application for nonlinear mechanical problems. The finite element method belongs to the family of weighted residual methods.

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

1. Make a guess (trial function) that has a number of unknown parameters. This is written as, for a mechanical problem with one unknown displacement field as $u(x)$,

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

where the functions $\varphi_i(x)$ often are polynomials. This is a displacement based formulation which is the standard approach in finite element formulation of mechanical problems. It will be temperature in the case of thermal problems. 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).

2. The trial function is inserted into the partial differential equation and the boundary conditions of the problem. The variant leading to a standard displacement based 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 be in our fictive case

$$L(\hat{u}) = R \neq 0 \qquad (3.2)$$
$$B_e(\hat{u}) \equiv 0 \text{ , } B_n(\hat{u}) = R_n \neq 0$$

where L denotes the differential equation of the problem and is defined over a 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.

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 is called a Galerkin method. This step generates the same number of equations as number of unknowns.

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

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

$$\boldsymbol{KU} = \boldsymbol{F}_{ext} \qquad (3.4)$$

where $\boldsymbol{K}$ is called a stiffness matrix in mechanical problems and $\boldsymbol{F}_{ext}$ is a load vector due to different kind of loadings including possible natural boundary conditions.

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

6. Derived quantities like strain and stress that are derivatives of the approximate solution has a larger error than the primary variables in $\boldsymbol{U}$.

Many textbooks formulate the finite element method for mechanical, elastic problems using the theorem of minimum potential energy. This theorem also requires that the essential boundary conditions of the problem are fulfilled like we introduced in step 3 above.

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

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

2. The trial function is set into the expression for the total potential energy, which is integrated over the domain like the first term in Eq. (3.3). This can be written in matrix form as

$$\Pi(\boldsymbol{U}) = \frac{1}{2}\boldsymbol{U}^T\boldsymbol{K}\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{F}_{ext} \tag{3.6}$$

3. The potential energy is stationary w.r.t to the parameters leading to

$$\boldsymbol{K}\boldsymbol{U} = \boldsymbol{F}_{ext} \tag{3.7}$$

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

6. Derived quantities like strain and stress that are obtained from derivatives of the approximate solution have a larger error than the primary variables in $\boldsymbol{U}$.

The difference between the approaches is in step 3. The energy method needs fewer manipulations at this step. However, it cannot be applied for nonlinear problems like those involving plasticity. Then the WRM approach can still be applied. The principle of *virtual power or work* is used in many textbooks to derive the finite element method for nonlinear mechanical problems. A comparison would show that it is the same as WRM.

Eq. (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

$$u_e(x) = \sum_{i=1}^{nnode} u_i N_i(x) = \boldsymbol{N}\boldsymbol{u} \tag{3.8}$$

where $\boldsymbol{N}$ is a matrix with interpolation ′functions, also often called shape functions as they determine the shape of the possible displacement field on element can describe. $\boldsymbol{u}$ is a vector with the nodal displacements of the element. *nnode* is the number of nodes in one 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 must be combined so that there is no mismatch between the displacement fields along common boundaries of elements. The most crucial step in the finite element modelling process is the choice of elements and the discretisation of the domain.

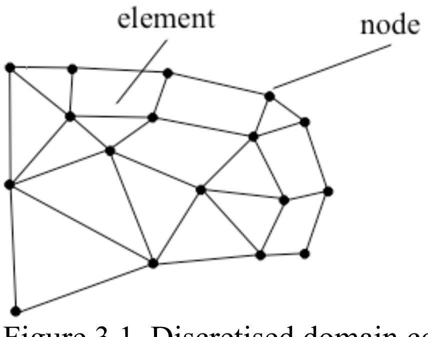


Figure 3.1. Discretised domain consisting of three and four node elements.

Different physics have different mathematical formulations but share some basic features. They are illustrated in Figure 3.2 for a static, mechanical problem. The equations are discussed later but we summarise them already here. The left side of the diagram are the kinematic variables describing the motion, u, and the gradient of it, i.e. the strain ε. Therefore, the matrix **B** contains derivatives of the interpolations functions **N**, i.e. the shape functions.

The constitutive equation, **E**, relates strain to stresses, for example Hooke's law. In other problems it is also common that some kind of gradient is related to some kind of flux. For example, the gradient of the temperature gives the heat flux in thermal problems. The model for this is Fourier's heat conduction law. The stresses are to be in equilibrium with applied forces. The line from stress, $\boldsymbol{\sigma}$, to the box symbolising the equilibrium equations is dashed. This means that this equation is approximated and that is why it is an integral. The equation is only fulfilled at the nodes with nodal equilibrium for the system written as

$$\mathbf{F}_{int} = \mathbf{F}_{ext} \quad (3.9)$$

This is a more general form than in Eq. (3.7).

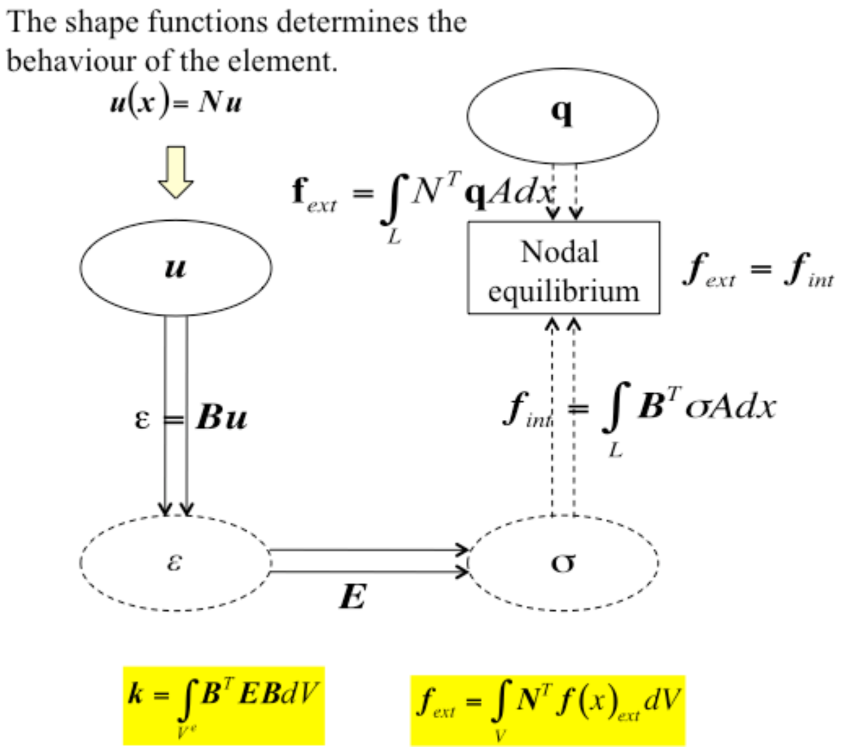

The shape functions determines the
behaviour of the element.
$u(x) = Nu$
q
$\mathbf{f}_{ext} = \int_L N^T \mathbf{q} A dx$
u
Nodal equilibrium
$f_{ext} = f_{int}$
$\varepsilon = Bu$
$f_{int} = \int_L B^T \sigma A dx$
ε
σ
E
$k = \int_{V^e} B^T E B dV$
$f_{ext} = \int_V N^T f(x)_{ext} dV$

Figure 3.2. A Tonti diagram illustrating the basic finite element relations in mechanics.

4 Weighted Residual Methods

The Weighted Residual Method is illustrated on a simple one-dimensional problem. 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;

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

$$I(\hat{u}) = R_I \neq 0 \tag{4.5}$$

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

The approximate solution can be structured so that;

i) $R \equiv 0$. Then it is called a boundary method.

ii) $R_B \equiv 0$. Then it is called an interior method This requirement may be violated for some boundary conditions reducing the efficiency[1] of the method.

iii) Else it is a mixed method.

Boundary element methods[2] are example of boundary methods. Green functions[3] 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

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

where $\varphi_i(x)$ are analytic functions called the trial functions. $u_0(x,t)$ must satisfy initial and boundary conditions as exact as possible. 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.

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

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

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 $N \to \infty$, 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

$$\lim_{N\to\infty} \int_\Omega \left\| \hat{u} - u_{exact} \right\|_2 dv = 0 \tag{4.9}$$

[1] 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.

[2] http://www.boundary-element-method.com/intro.htm

[3] http://en.wikipedia.org/wiki/Green%27s_function

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

$$w_k = \begin{cases} 1 \text{ in } D_k \\ 0 \text{ else} \end{cases} \quad \text{for } k = 1,\ldots N \tag{4.10}$$

One example of this is the finite-volume method[4]. Each element is surrounding its associated node. Conservation equations then relates changes within this volume with fluxes over its boundaries.

4.2 Collocation method

The weight function is given by

$$w_k = \delta\left(x - x_k\right) \;\text{ for }\; k = 1,\ldots N \tag{4.11}$$

where δ is the Dirac delta function[5]. Thus the residual is forced to be zero at specific locations.

4.3 Least-squares method

The weight function is given by

$$w_k = \frac{\partial R}{\partial a_k} \quad \text{for } k = 1,\ldots N \tag{4.12}$$

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

$$\int_\Omega \frac{\partial R}{\partial a_k} R dv = 0 \text{ for } k = 1,\ldots N \tag{4.13}$$

This in turn is the stationary value of

$$\Pi\left(a_1, a_2 \ldots a_N\right) = \int_\Omega R^2 dv \tag{4.14}$$

thereby motivating the name of the method.

4.4 Method of moments

The weight function is in this case given by

$$w_k = x^k \quad \text{for } k = 1,\ldots N - 1 \tag{4.15}$$

4.5 Galerkin and Ritz methods

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

$$w_k = \varphi_k\left(x\right) \;\text{ for }\; k = 1,\ldots N \tag{4.16}$$

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 method is the same as the *principle of virtual work or power*[6] used in mechanics when formulating the finite element method. The weight functions correspond to virtual

[4] http://en.wikipedia.org/wiki/Finite_volume_method

[5] http://en.wikipedia.org/wiki/Dirac_delta_function

displacements or velocities in this approach. For elastic problems, it also corresponds to the *principle of minimum total energy*[7,8]. 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. However, it is not valid in cases like plasticity. Then the principle of virtual work is used.

4.5.1 Relation between the Galerkin and Ritz methods

Thus the Galerkin method is more general than Ritz method, in the same way as the principle of virtual work is more general than the principle of minimum total potential energy.

The relation between the Galerkin method and Ritz method can be described as follows. Assume that u is the solution to the differential equation

$$A(u) = f_{\text{ext}} \tag{4.17}$$

where A is a positive definite operator. This property means that

$$\int_\Omega uA(u)dv > 0 \text{ for all } u \neq 0 \tag{4.18}$$

Then the solution to Eq. (4.17) can be shown to be equivalent to finding the minimum of the functional

$$\Pi(u) = \frac{1}{2}\int_\Omega uA(u)dv - \int_\Omega uf_{ext}dv \tag{4.19}$$

An approximate solution like in Eq. (4.7) is used but now it only fulfils the essential boundary conditions. Assuming that this fulfilment can be done by fixing appropriate coefficients a_i leads to

$$\Pi(\hat{u}) = \frac{1}{2}\int_\Omega \sum_{k=1}^{N} a_k(t)\varphi_k(x)\sum_{i=1}^{N} a_i(t)A(\varphi_i(x))dv - \int_\Omega \sum_{k=1}^{N} a_k(t)\varphi_k(x)f_{ext}dv \tag{4.20}$$

This can be written in matrix form as

$$\Pi(\hat{u}) = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{Ka} - \boldsymbol{aF}_{ext} \tag{4.21}$$

where the coefficients of the matrix and vector are

$$K_{ki} = \int_\Omega \varphi_k(x)A(\varphi_i(x))dv \tag{4.22}$$

and

$$F_{ext,k} = \int_\Omega \varphi_k(x)f_{ext}dv \tag{4.23}$$

The best choice of parameters is the set that minimise this functional. A condition for this is

$$\frac{\partial\Pi(\hat{u})}{\partial a_k} = 0 \quad \text{for } k = 1,...N \tag{4.24}$$

This leads to

$$\boldsymbol{Ka} = \boldsymbol{F}_{ext} \tag{4.25}$$

This is the same as for the Galerkin method as can be seen in section 4.7.6. The convergence properties for the Ritz method states that increasing N makes the functional Π go towards the

[6] http://en.wikipedia.org/wiki/Virtual_work

[7] http://en.wikiversity.org/wiki/Introduction_to_Elasticity/Principle_of_minimum_potential_energy

[8] http://en.wikipedia.org/wiki/Minimum_total_potential_energy_principle

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.

4.6 Petrov-Galerkin method

The weight function is represented by

$$w_k = P_k(x) \text{ for } k = 1,...N-1 \tag{4.26}$$

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.

4.7 Comparison of WRM methods

4.7.1 Problem definition and exact solution

The different weighted residual methods will be applied on the problem

$$L(u) = a\frac{d^2u}{dx^2} + u - f_{ext} = A(u) - f_{ext} = 0 \quad \text{where } x \in [0,1] \tag{4.27}$$

It is assumed that the function *u* is written in non-dimensional form, ie it does not have any dimensions.

$$B_e(0) = u(0) - 1 = 0 \tag{4.28}$$

$$B_n(0) = b\left.\frac{du}{dx}\right|_{x=1} - \frac{1}{\cos(1)} = 0 \text{ or } B_n(1) = b\left.\frac{du}{dx}\right|_{x=1} - 1 = 0 \tag{4.29}$$

We will limit our discussion to the particular case of a=1 [m^2] and b=1 [m] and $f_{ext} = 1$. 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. (4.29) 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 (4.27)-(4.29) is

$$u_{exact} = \frac{\sin(x)}{\cos(1)} + 1 \tag{4.30}$$

We choose an approximate solution given by

$$\hat{u} = \sum_{j=1}^{N} a_j\varphi_j = \sum_{j=1}^{N} a_j x^{j-1} \tag{4.31}$$

This gives the residuals

$$L(\hat{u}) = \frac{d^2\hat{u}}{dx^2} + \hat{u} - 1 = \sum_{j=3}^{N} a_j (j-1)(j-2)x^{j-3} + \sum_{j=1}^{N} a_j x^{j-1} - 1 = R \tag{4.32}$$

$$B_e(\hat{u}) = \hat{u}(0) - 1 = R_e \tag{4.33}$$

$$B_n(\hat{u}) = \hat{u}'(0) - \frac{1}{\cos(1)} = R_n \text{ or } B_n(\hat{u}) = \hat{u}'(1) - 1 = R_n \tag{4.34}$$

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

$$\hat{u}(0) = a_1 = 1 \qquad (4.35)$$

The natural boundary condition can also be used to impose conditions on the parameters. Eq. (4.29) gives directly

$$\left.\frac{d\hat{u}}{dx}\right|_{x=0} = a_2 = \frac{1}{\cos(1)} \qquad (4.36)$$

4.7.2 Subdomain example

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

$$\int_{x_{k-1}}^{x_k} R dv = 0 \quad \text{for } k = 1,...N \qquad (4.37)$$

where

$$x_k = \frac{k}{N} = \text{k}\Delta\text{x} \quad \text{for } k = 1,...N \qquad (4.38)$$

This can be integrated giving, for each *k*,

$$\sum_{j=3}^{N} a_j (j-1)\left(x_k^{j-2} - x_{k-1}^{j-2}\right) + \sum_{j=1}^{N} \frac{a_j}{j}\left(x_k^j - x_{k-1}^j\right) = \Delta x \qquad (4.39)$$

This leads to a system of equations with

$$K_{kj} a_j = F_{ext,k}{}^{9} \qquad (4.40)$$

where the right hand side is

$$F_{ext,k} = 1/N \text{ for all } k$$

and the matrix on the left hand side becomes

$$K_{kj} = \frac{x_k^j - x_{k-1}^j}{j} \text{and if j >2 } + (j-1)\left(x_k^{j-2} - x_{k-1}^{j-2}\right)$$

The table below shows the results for the subdomain method and its condition number and the implementation is shown in Table 4.2. A high condition number and indicates that the solution is sensitive to round off error. It may even be impossible to invert the matrix.

Table 4.1. L2-error and condition number of matrix for different subdomain solutions.

Number of terms (*N*)	Error	Condition number	Number of terms (*N*)	Error	Condition number
3	1.05	1.0000	9	1.2e-6	7.7e5
4	0.18	27.8	10	3.3e-8	5.3e6
5	0.04	2.9e2	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

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

Table 4.2. Excerpt from Matlab code for the subdomain method.

```
for k=1:N
    F(k)=dx;
    xk=k*dx;
    xk_1=xk-dx;
    for j=1:N
        K(k,j)=(xk^j-xk_1^j)/j;
        if j>2
            K(k,j)=K(k,j)+(j-1)*(xk^(j-2)-xk_1^(j-2));
        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=F-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)=[];
```

4.7.3 Collocation example

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 domain of equal length. Thus *N* locations cause these points to be at

$$x_k = \frac{k-0.5}{N} = \mathrm{k}\Delta\mathrm{x} \quad \text{for } k = 1,...N \tag{4.41}$$

Then Eq. (4.8) becomes

$$\sum_{j=3}^{N} a_j (j-1)(j-2) x_k^{j-3} + \sum_{j=1}^{N} a_j x_k^{j-1} - 1 \ = 0 \quad \text{for } k = 1,...N \tag{4.42}$$

This leads to a system of equations with

$$K_{kj} a_j = F_{ext,k} \tag{4.43}$$

where the right hand side is

$$F_k = 1 \text{ for all } k$$

And the matrix on the left hand side becomes

$$K_{kj} = x_k^{j-1} \text{ and if j >2 } + x_k^{j-3}$$

The table below shows the results for the subdomain method and the implementation is shown in Table 4.4

Table 4.3. L2-error and condition number of matrix for collocation method.

Number of terms (*N*)	Error	Condition number	Number of terms (*N*)	Error	Condition number
3	1.0536	1.0	7	3.4e-4	1.5e4
4	0.1793	27.7	8	1.4e-5	1.0e5
5	0.0420	283.	9	1.3e-6	7.5e6
6	0.0026	2.e3	10	3.8e-8	5.3e7

Table 4.4. Excerpt from Matlab code for the point collocation method.

```
for k=1:N
    F(k)=1;
    xk=(k-0.5)*dx;
    for j=1:N
        K(k,j)=xk^(j-1);
        if j>2
            K(k,j)=K(k,j)+(j-2)*(j-1)*xk^(j-3);
        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=F-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)=[];
```

4.7.4 Least squares method

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

$$\int_0^1 \left((k-1)(k-2)x^{k-3}+x^{k-1}\right)\left[\sum_{j=3}^{N} a_j (j-1)(j-2)x^{j-3}+\sum_{j=1}^{N} a_j x^{j-1}-1\right]dx=0 \tag{4.44}$$

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

$$K_{kj}a_j = F_{ext,k} \tag{4.45}$$

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

$$F_{ext,k}=\frac{1}{k}\text{ and if k>2 }+(k-1)$$

The matrix, that is symmetric, becomes

$$K_{kj}=\frac{1}{j+k-1}\text{ and if k > 2 }+\frac{(k-1)(k-2)}{j+k-3}$$
$$\text{and if j >2 }+\frac{(j-1)(j-2)}{k+j-3}\text{ and if k,j >2 }+\frac{(j-1)(j-2)(k-1)(k-2)}{k+j-5}$$

The results are shown in Table 4.5 and the implementation in Table 4.6. 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 4.5. L2-error and condition number of matrix for least squares solution.

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Reciprocal condition number
3	0.56	1.0000	7	6.3e-7	4.5e5
4	0.009	37.4	8	1.0e-8	1.3e6
5	6.4e-4	736.	9	5.46e-10	4.0e8
6	1.1e-5	1.7e4	10	3.5e-9	1.1e10

Table 4.6. Excerpt from Matlab code for the least squares method.

```
for k=1:N
    k1k2=(k-1)*(k-2);
    F(k)=1/k;
    if k>2
        F(k)=F(k)+k-1;
    end
    for j=1:N
        j1j2=(j-1)*(j-2);
        K(k,j)=1/(j+k-1);
        if k>2 && j> 2
            K(k,j)=K(k,j)+(j1j2+k1k2)/(j+k-3)+j1j2*k1k2/(k+j-5);
        elseif k>2
            K(k,j)=K(k,j)+k1k2/(j+k-3);
        elseif j>2
            K(k,j)=K(k,j)+k1k2/(j+k-3);
        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=F-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)=[];
```

4.7.5 Method of moments example

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

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

Further elaboration gives leads to a system of equations with

$$K_{kj} a_j = F_{ext,k} \tag{4.47}$$

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

$$F_{ext,k} = \frac{1}{k+1}$$

The matrix on the left hand side becomes

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

The results are shown in Table 4.7 and the implementation in Table 4.8. The error increases when *N*=10 and fails completely at *N*=13.

Table 4.7. L2-error and condition number of matrix for method of moments solution.

Number of terms (*N*)	Error	Condition number	Number of terms (*N*)	Error	Condition number
3	0.99	1.0000	7	1.3e-4	2.9e6
4	0.142	236.	8	3.8e-6	1.1e7
5	0.026	1.7e4	9	3.28e-7	4.2e8

6	1.3e-3	8.0e5	10	2.63e-6	1.4e9

Table 4.8. Excerpt from Matlab code for method of moments.

```
for k=1:N
    F(k)=1/(k+1);
    for j=1:N
        K(k,j)=1/(j+k);
        if j>2
            K(k,j)=K(k,j)+(j-1)*(j-2)/(j+k-2);
        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=F-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)=[];
```

4.7.6 Galerkin example

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

$$\int_0^1 x^{k-1} A\left(\sum_{j=1}^{N} a_j x^{j-1}\right) - 1\cdot x^{k-1} dx = 0 \qquad (4.48)$$

The notation A was introduced for comparison with the Ritz method discussed in the context of Galerkin method in section 4.5. The operator A is on the approximate solution is $A(\hat{u}) = \frac{d^2\hat{u}}{dx^2} + \hat{u}$. This leads to

$$\sum_{j=3}^{N} a_j \frac{(j-1)(j-2)}{k+j-3} 1^{k+j-3} + \sum_{j=1}^{N} \frac{a_j}{k+j-1} 1^{k+j-1} - \frac{1^k}{k} = 0 \qquad (4.49)$$

Further elaboration gives leads to a system of equations with

$$K_{kj} a_j = F_{ext,k} \qquad (4.50)$$

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

$$F_{ext,k} = \frac{1}{k}$$

The matrix on the left hand side becomes

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

The results shown below have the same accuracy as the method of moments due to the special choice of trial/weight functions. The results are shown in Table 4.7 and the implementation in Table 4.8.

Table 4.9. L2-error and condition number of matrix for Galerkin method.

Number of terms (N)	Error	Condition number	Number of terms (N)	Error	Condition number
3	0.99	1.0000	7	1.3e-4	2.9e7
4	0.142	236.	8	3.8e-6	1.1e8
5	0.026	1.7e5	9	3.28e-7	4.2e8
6	1.3e-3	8.0e6	10	2.63e-6	1.4e9

Table 4.10. Excerpt from Matlab code for Galerkin's method.

```matlab
for k=1:N
    F(k)=1/(k+1);
    for j=1:N
        K(k,j)=1/(j+k);
        if j>2
            K(k,j)=K(k,j)+(j-1)*(j-2)/(j+k-2);
        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=F-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)=[];
```

4.7.7 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 [3] with subjective comparisons of the some of the methods.

Table 4.11. Subjective comparisons of different weighted residual methods, from Fletcher [3].

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.

5 Classical and computational Galerkin methods

The classical Galerkin methods were applied before computers were commonplace. Thus there was a need to obtain high accuracy with few unknowns. The method used global test functions. The use of orthogonal[10] test functions further reduced the calculations needed. Naturally, the use of global functions also made it difficult to solve problems with irregular boundaries. 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 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 lead to that the system of equations that is to be solved in order to determine the coefficients a_j will be 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 shown next are based on the use of local trial functions instead. Increasing the number of coefficients is done by increase the number of domains, elements, they are defined over. The way these domains, elements, are described is also a key to one of the strong points of Finite Element Methods, their ability to solve problems with complex boundaries.

6 Finite Element Methods

A Finite Element Method (FEM) solution of the same problem as earlier will be formulated with the use of the Galerkin method but now allowing the trial solution to approximate the natural boundary condition. 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 in the spirit of the Finite Element Method, will be used now. Two approaches for the FE-formulation 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 way to implement the method. Iso-parametric formulation will be introduced in this context.

The problem given in section 4.7 will be solved but the natural boundary condition will be prescribed at the right end and will be approximated in the solution[11]. The equations are repeated below.

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

$$B_e(0) = u(0) - 1 = 0 \tag{6.2}$$

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

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

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

[10] http://en.wikipedia.org/wiki/Orthogonal_functions $\int \varphi_k \varphi_k dV = 0$ if $k \neq j$

[11] We cannot set it at x=0 in the current approach as we will later set the weight function to zero where we have the essential boundary condition, which also is at x=0.

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

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

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

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

There is a second derivative on the trial functions but no on the weight functions. We perform a partial integration[12] leading to

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

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

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

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

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

or

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

If it was 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 earlier before the finite element version is shown in the next section. The approximate solution, and corresponding weight functions, are

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

$$\hat{u} = \sum_{j=1}^{N} a_j \varphi_j = \sum_{j=1}^{N} a_j x^{j-1} \tag{6.11}$$

Inserted into Eq. (6.10) gives

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

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. (4.50) where *j* starts with 3 (j>2) in the summation. This reduction is due to the partial integration performed.

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

$$\sum_{j=1}^{N} a_j \frac{(k-1)(j-1)}{k+j-3} - \sum_{j=1}^{N} \frac{a_j}{k+j-1} = 1 - \frac{1}{k} \quad \text{for } k = 1,\ldots N \tag{6.12}$$

A comparison with Eq. (4.50) shows that the problem is now symmetric with respect to indexes *j* and *k*. We has now

$$K_{kj} a_j = F_{ext,k} \tag{6.13}$$

with

$$F_{ext,k} = 1 - \frac{1}{k}$$

The matrix on the left hand side becomes

$$K_{kj} = -\frac{1}{j+k-1} \text{ and if } j,k > 1 \; + \frac{(k-1)(j-1)}{k+j-3}$$

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 if the error in the natural boundary conditions would have been included in the Galerkin method in Eq. (4.48) then the errors in Table 4.9 and Table 6.1 would have been the same as the partial integration only makes the matrix symmetric but will bring in any new assumptions. The error increases from *N*=9 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	0.072	18.3	7	2.6e-7	8.5e6
4	0.002	375.	8	4.8e-9	2.7e8
5	1.8e-4	1.0e4	9	4.3e-8	8.4e9
6	4.1e-6	2.8e5	10	9.1e-7	2.5e11

Table 6.2. Excerpt from Matlab code for Ritz method case, i.e. a symmetric variant of the Galerkin's method.

```
for k=1:N
    F(k)=1-1/k;
    for j=1:N
        K(k,j)=-1/(j+k-1);
        if j>1 && k >1
            K(k,j)=K(k,j)+(j-1)*(k-1)/(j+k-3);
        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=F-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

$$\varphi_j = \begin{cases} \dfrac{x\text{-}x_{j\text{-}1}}{x_j - x_{j-1}} & \text{for } x_{j-1} \le x \le x_j \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j} & \text{for } x_j \le x \le x_{j+1} \end{cases} \tag{6.14}$$

where

$$\Delta x = x_j - x_{j-1} = x_{j+1} - x_j \tag{6.15}$$

is the length of each segment and assumed constant for simplicity in the following.

A typical function is shown in Figure 6.1. Note that the very first node (j=1) has only the right half of the function and the final node (j=N) has only the left part of the function.

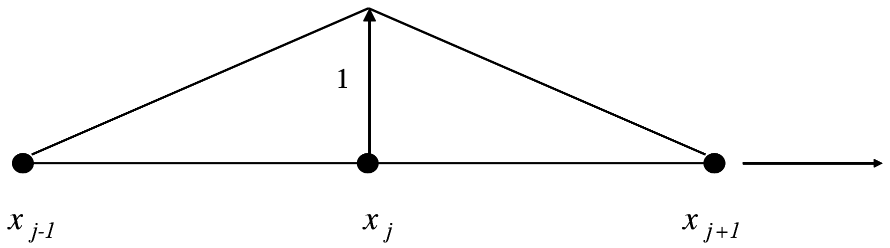


Figure 6.1. Finite element type of trial functions or shape functions with local base.

The approximate solution is then written as

$$\hat{u} = \sum_{j=1}^{N} u_j \varphi_j \tag{6.16}$$

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

$$\varphi_j(x_k) = \delta_{ij} \tag{6.17}$$

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

$$\hat{u}(x_j) = u_j$$

This 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 particular 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 finite element method giving its ultimate flexibility. Now – back on track!

Insertion of Eq. (6.14)into Eq. (6.10) gives

$$\int_0^1 \frac{d\varphi_k}{dx}\sum_{j=1}^{N} u_j \frac{d\varphi_j}{dx}dx - \int_0^1 \varphi_k\left(\sum_{j=1}^{N} u_j\varphi_j - 1\right)dx = [\varphi_k \cdot 1]_{x=1} \quad \text{for } k = 1,...N \tag{6.18}$$

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 gray area in Figure 6.2.

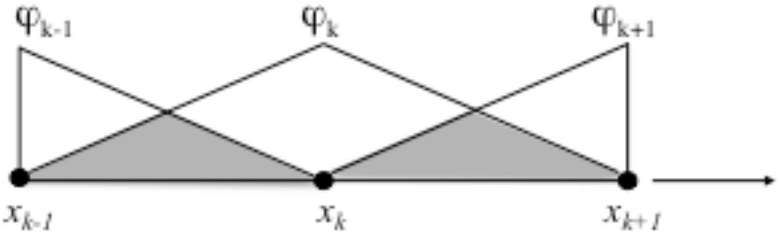


Figure 6.2. Overlap between neighbours where the trial functions contribute to integrals to be evaluated. Thus equation k will have contributions from three trial functions *j*=*k*-1, *j*=*k* and *j*=*k*+1.

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

$$\int_{x_{k-1}}^{x_{k+1}} \frac{d\varphi_k}{dx}\sum_{j=k-1}^{k+1} u_j \frac{d\varphi_j}{dx}dx - \int_{x_{k-1}}^{x_{k+1}} \varphi_k\left(\sum_{j=k-1}^{k+1} u_j\varphi_j - 1\right)dx = 0 \tag{6.19}$$

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

$$\int_0^{x_1} \frac{d\varphi_k}{dx}\sum_{j=k}^{k} u_j \frac{d\varphi_j}{dx}dx - \int_0^{x_1} \varphi_k\left(\sum_{j=k}^{k} u_j\varphi_j - 1\right)dx = 0 \text{ for } k = 1 \tag{6.20}$$

$$\int_{x_{N-1}}^{x_N} \frac{d\varphi_k}{dx}\sum_{j=N}^{N} u_j \frac{d\varphi_j}{dx}dx - \int_{x_{N-1}}^{x_N} \varphi_k\left(\sum_{j=N}^{N} u_j\varphi_j - 1\right)dx = [\varphi_N \cdot 1]_{x=1} \text{ for } k = N \tag{6.21}$$

The derivatives $\frac{d\varphi_j}{dx}$ and $\frac{d\varphi_k}{dx}$ are the slopes of the functions with the values $\pm\frac{1}{\Delta x}$ where Δx is assumed to be constant as we limit to constant distance between the points. The integrals with combinations of φ_j are polynomials that can be integrated analytically. This gives

$$K_{kj}a_j = F_{ext,k} \tag{6.22}$$

with

$$F_{ext,k} = -\Delta x$$

with the exception of

$$F_{ext,k} = -\frac{\Delta x}{2} \quad \text{for } k = 1$$

$$F_{ext,k} = 1 - \frac{\Delta x}{2} \quad \text{for } k = N$$

The matrix on the left hand side becomes

$$K_{kj} = \frac{2}{\Delta x} \ -\frac{2}{3}\Delta x \quad \text{for } k = j$$

$$K_{kj} = -\frac{1}{\Delta x} - \frac{1}{6}\Delta x \quad \text{for } k = j+1 \text{ or } k = j-1$$

with the exception of

$$K_{kj} = \frac{1}{\Delta x} - \frac{1}{3}\Delta x \quad \text{for } k = j = 1 \text{ or } k = j = N$$

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 6.3. 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 2.6e-7 is when using *N*=10000.

Table 6.3. 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	0.77	1.0000	6	0.039	99.8
3	0.232	13.9	7	0.027	140.
4	0.107	39.1	8	0.020	188.
5	0.061	66.1	9	0.015	242

Table 6.4. Excerpt from Matlab code for the Finite Element Method.

```
F(1)=-dx*0.5;
F(N)=1-dx*0.5;
invdx=1/dx;
K(1,1)=invdx-dx/3;
K(N,N)=K(1,1);
kdiag=2*K(1,1);
koffdiag=-1/dx-dx/6;
K(1,2)=koffdiag;
K(2,1)=koffdiag;
for k=2:N-1
    F(k)=-dx;
    K(k,k)=kdiag;
    K(k,k+1)=koffdiag;
    K(k+1,k)=koffdiag;
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=F-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)=[];
```

6.3 Element based trial and weight functions

We will make some observations on the previous example before we make a more efficient variant of the finite element formulation.

The previous example showed that the matrix **K** was banded. It was filled along the diagonal and the nearest off-diagonals. The general property of finite element matrices is that they are not full. However, they need not have the nice structure as in the example above. The number at each position (k,j) can be interpreted in terms of the physics of the problem. One obvious interpretation is that this number give the contribution from the j:te degree of freedom, u_j, to the k:th 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 $u(x)$ 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) whereas the right end has a given flux. Therefore, it enters naturally into the equilibrium equation as a term, see the right hand side of Eq. (6.22). That is why it is called a natural boundary condition.

Different ordering of the node numbers, or rather equation numbers[13], 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 K. The two examples in Figure 6.3 fill the matrix **K** differently. The top example in the figure gives

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

$$\boldsymbol{K} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & x & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \tag{6.23}$$

whereas the lower case gives

$$\boldsymbol{K} = \begin{bmatrix} x & x & 0 & 0 \\ x & x & 0 & x \\ 0 & 0 & x & x \\ 0 & x & x & x \end{bmatrix} \tag{6.24}$$

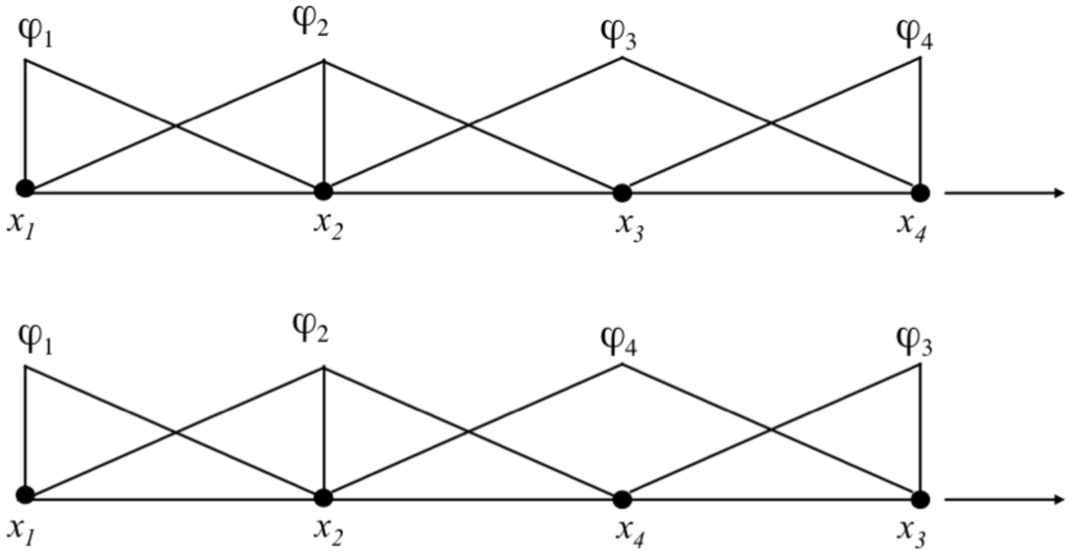


Figure 6.3. Two examples of numbering of nodes.

All coefficients in the matrix and vector of Eq. (6.22) that correspond to nodes in the interior of the domain obtained contributions from an integral over the domains surrounding this node. One can note that we then 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 common 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. The previous FE-example gives $u(x)$ from linear interpolation between the nodes at the left and right end of the segment.

Based on this observation we make the element the primary entity and describe the total field by splitting the domain into elements and writing

$$\hat{u}(x) = \sum_{e=1}^{Nelem} u^e(x) \tag{6.25}$$

Nelem is the number of elements used to solve the problem. The field within an element is written as

$$u^e(x) = \sum_{i=1}^{Nnode} N_i u_i = \boldsymbol{N}\boldsymbol{u} \tag{6.26}$$

where *Nnode* is the number of nodes associated with the element. Now we change the notation once more for the trial functions as we switch to element description. Eq. (6.34) can be written for our earlier example as

$$N(x)u = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{x_2 - x}{l^e} & \frac{x - x_1}{l^e} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{6.27}$$

where $l^e = x_2 - x_1$ is the element length. This is the same as our *dx* used earlier. **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 6.4. 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 letters 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).

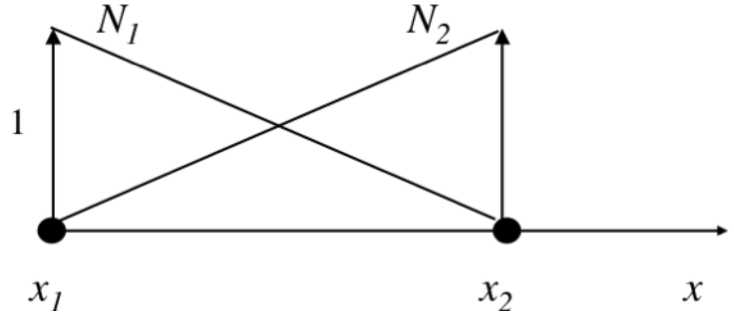


Figure 6.4. Two node element and its linear shape functions N_1 and N_2.

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 6.5.

$$u(s(x)) = N(s)u = \begin{bmatrix} \frac{1-s}{2} & \frac{1+s}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{6.28}$$

This variable change can be described in the same way

$$x(s) = N(s)x = \begin{bmatrix} \frac{1-s}{2} & \frac{1+s}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6.29}$$

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.

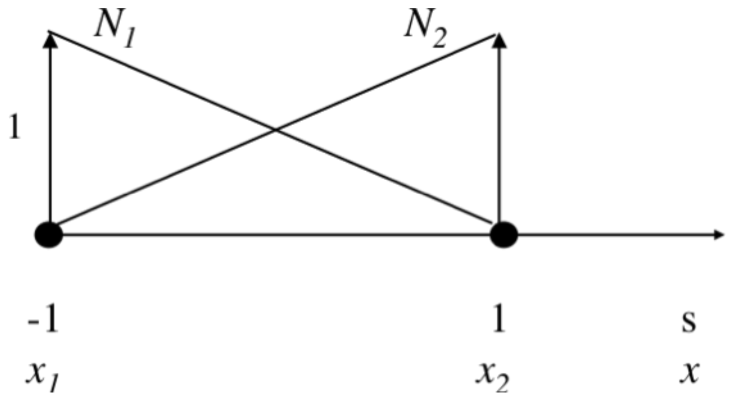


Figure 6.5. Two node element and its linear shape functions N_1 and N_2 with a local coordinate system, s.

The integrals in Eq. (6.19) will now be integrated elementwise as

$$\int_{x_1}^{x_2} \left(\frac{d\boldsymbol{N}}{dx}\right)^T \frac{d\boldsymbol{N}}{dx}\boldsymbol{u}dx - \int_{x_1}^{x_2} \boldsymbol{N}^T\boldsymbol{N}\boldsymbol{u} - \boldsymbol{N}^T \cdot 1dx \tag{6.30}$$

$\frac{d\boldsymbol{N}}{dx}$ has the derivatives of both shape functions of the element, one for the left node and one for the right node. Thus Eq. (6.30) includes the contribution from the trial <u>and</u> weight functions of these two nodes to the overall system of equations needed. We introduce a specific notation for the $\frac{d\boldsymbol{N}}{dx}$ as this kind of expression is common. We denote it by

$$\boldsymbol{B} = \frac{d\boldsymbol{N}}{dx} \tag{6.31}$$

We can also move the u-terms outside the integrals as the nodal values do not depend on the coordinates. Thus we have

$$\int_{x_1}^{x_2} \boldsymbol{B}^T\boldsymbol{B}dx\boldsymbol{u} - \int_{x_1}^{x_2} \boldsymbol{N}^T\boldsymbol{N}dx\boldsymbol{u} + \int_{x_1}^{x_2} \boldsymbol{N}^T \cdot 1dx \tag{6.32}$$

Notice that we contribute to two *k*:s at the same time as $\boldsymbol{B}^T$ and $\boldsymbol{N}^T$ 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

$$\frac{d\boldsymbol{N}(s(x))}{dx} = \frac{d\boldsymbol{N}}{ds}\frac{ds}{dx} \tag{6.33}$$

Eq. (6.29) gives

$$\frac{dx}{ds} = J = \frac{d\boldsymbol{N}}{ds}\boldsymbol{x} = \begin{bmatrix} \frac{-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{x_2 - x_1}{2} = \frac{l^e}{2} \tag{6.34}$$

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

$$\frac{ds}{dx}=J^{-1}=\frac{2}{l^e} \tag{6.35}$$

Now we can write

$$\boldsymbol{B}=\frac{d\boldsymbol{N}}{dx}=\frac{d\boldsymbol{N}}{ds}J^{-1}=\begin{bmatrix}\frac{-1}{2} & \frac{1}{2}\end{bmatrix}\frac{2}{l^e}=\begin{bmatrix}\frac{-1}{l^e} & \frac{1}{l^e}\end{bmatrix} \tag{6.36}$$

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. (6.32) in the local coordinate system. We get

$$\int_{-1}^{1}\boldsymbol{B}^T\boldsymbol{B}Jds\boldsymbol{u}-\int_{-1}^{1}\boldsymbol{N}^T\boldsymbol{N}Jds\boldsymbol{u}+\int_{-1}^{1}\boldsymbol{N}^T\cdot 1Jds \tag{6.37}$$

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

$$\boldsymbol{k}_1\boldsymbol{u}-\boldsymbol{k}_2\boldsymbol{u}+\boldsymbol{f}_{ext}=\boldsymbol{k}\boldsymbol{u}+\boldsymbol{f}_{ext} \tag{6.38}$$

where

$$\boldsymbol{k}_1=\int_{-1}^{1}\begin{bmatrix}\frac{-1}{l^e}\\ \frac{1}{l^e}\end{bmatrix}\begin{bmatrix}\frac{-1}{l^e} & \frac{1}{l^e}\end{bmatrix}\frac{l^e}{2}ds=\int_{-1}^{1}\begin{bmatrix}\frac{1}{l^{e^2}} & -\frac{1}{l^{e^2}}\\ -\frac{1}{l^{e^2}} & \frac{1}{l^{e^2}}\end{bmatrix}\frac{l^e}{2}ds=\begin{bmatrix}\frac{1}{l^e} & -\frac{1}{l^e}\\ -\frac{1}{l^e} & \frac{1}{l^e}\end{bmatrix}$$

$$\boldsymbol{k}_2=\int_{-1}^{1}\begin{bmatrix}\frac{1-s}{2}\\ \frac{1+s}{2}\end{bmatrix}\begin{bmatrix}\frac{1-s}{2} & \frac{1+s}{2}\end{bmatrix}\frac{l^e}{2}ds=\int_{-1}^{1}\begin{bmatrix}\frac{(1-s)^2}{4} & \frac{1-s^2}{4}\\ \frac{1-s^2}{4} & \frac{(1+s)^2}{4}\end{bmatrix}\frac{l^e}{2}ds=\begin{bmatrix}\frac{l^e}{3} & \frac{l^e}{6}\\ \frac{l^e}{6} & \frac{l^e}{3}\end{bmatrix}$$

Giving

$$\boldsymbol{k}=\begin{bmatrix}\frac{1}{l^e} & -\frac{1}{l^e}\\ -\frac{1}{l^e} & \frac{1}{l^e}\end{bmatrix}+\begin{bmatrix}\frac{l^e}{3} & \frac{l^e}{6}\\ \frac{l^e}{6} & \frac{l^e}{3}\end{bmatrix} \tag{6.39}$$

$$\boldsymbol{f}_{ext}=\int_{-1}^{1}\boldsymbol{N}^T\cdot 1Jds=\int_{-1}^{1}\begin{bmatrix}\frac{1-s}{2}\\ \frac{1+s}{2}\end{bmatrix}\frac{l^e}{2}ds=\begin{bmatrix}\frac{l^e}{2}\\ \frac{l^e}{2}\end{bmatrix} \tag{6.40}$$

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 *j,k* our nodes correspond to. We will illustrate this in the Box below by making an assembly of the using the upper model in Figure 6.3. This is demonstrated for the lower model in Figure 6.3 in Box 6.2 but in a way that is another step towards what is done in an actual FE-software.

Box 6.1. Assembly procedure.

Logic for element assembly

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

Element 1 has node number 1 as local node number 1 and node number 2 as right node. Thus we add the matrix **k** according to Eq (6.39) to **K** and vector $\boldsymbol{f}_{ext}$ to $\boldsymbol{F}_{ext}$. Then we get

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

Notice the change in sign as we had $\boldsymbol{f}_{ext}$ on left hand side in Eq. (6.39) but $\boldsymbol{F}_{ext}$ is on the right hand side.

Element 2 has node number 2 as left node 1 and node number 3 as local node number 2. Thus we add the matrix **k** according to Eq. (6.39) to **K** and vector $\boldsymbol{f}_{ext}$ to $\boldsymbol{F}_{ext}$. Then we get

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

Element 3 has node number 3 as left node 1 and node number 4 as right node. Thus we add the matrix **k** according to Eq. (6.39) to **K** and vector $\boldsymbol{f}_{ext}$ to $\boldsymbol{F}_{ext}$. Then we get

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

This is the same as would be obtained using the formula in Eq. (6.22)

Box 6.2. Automated assembly procedure.

Logic for element assembly

Zero global matrix **K**, a 4x4 matrix, and global vector $\boldsymbol{F}_{ext}$, 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

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

This is used to automatic 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

$$\begin{array}{cc} & \begin{bmatrix} 1 & \qquad & 2 \end{bmatrix} \\ \boldsymbol{k} = & \begin{bmatrix} \frac{1}{l^e}+\frac{l^e}{3} & -\frac{1}{l^e}+\frac{l^e}{6} \\ -\frac{1}{l^e}+\frac{l^e}{6} & \frac{1}{l^e}+\frac{l^e}{3} \end{bmatrix}\begin{bmatrix} 1 \\ \\ 2 \end{bmatrix} \end{array} \quad \boldsymbol{f}_{ext} = \begin{bmatrix} \frac{l^e}{2} \\ \frac{l^e}{2} \end{bmatrix}\begin{bmatrix} 1 \\ \\ 2 \end{bmatrix}$$

Then we get

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

Notice the change in sign as we had $\boldsymbol{f}_{ext}$ on left hand side in Eq. (6.39) but $\boldsymbol{F}_{ext}$ is on the right hand side.

Element 2

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

with

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

Giving

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

Element 3

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

with

$$\begin{bmatrix} 4 & & 3 \end{bmatrix}$$

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

Giving

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

7 Numerical integration

The previous chapter brought us to isoparametric element but only in the one-dimensional context. Next chapter generalises 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[14].

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

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

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[15]. 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*, and number of integration points is

$$n = 2nint \text{ - } 1 \tag{7.2}$$

Table 7.1. Gauss integration rule.

Nint	Integration points	weights
1	0	2
2	$-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$	1,1

[14] http://en.wikipedia.org/wiki/Numerical_integration

[15] http://en.wikipedia.org/wiki/Gaussian_quadrature

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

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

The needed two point rule gives

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

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

$$I = \int_{-1}^{1}\int_{-1}^{1} f\,dsdt = \sum_{i=1}^{nint}\sum_{j=1}^{nint} f\left(s_i, t_i\right) w_i w_j \tag{7.5}$$

or

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f\,dsdtdz = \sum_{i=1}^{nint}\sum_{j=1}^{nint}\sum_{k=1}^{nint} f\left(s_i, t_i, z_k\right) w_i w_j w_k \tag{7.6}$$

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 0 noted 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 function[16]. However, this is provided the integrals are integrated exactly. This opens up 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 8.2.

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 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 did not discover any strain due to these modes. This must be avoided or prevented.

[16] It is the total potential energy in case of elasticity problems.

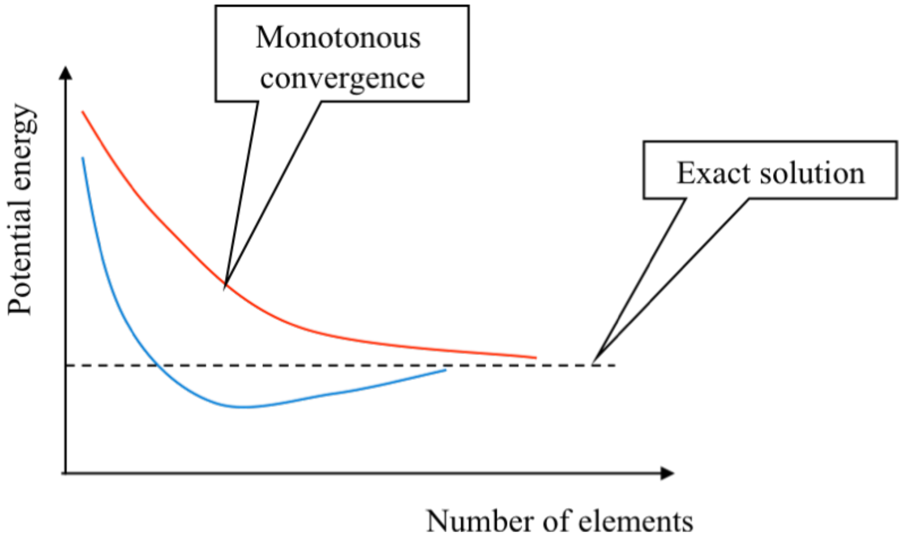


Figure 7.1. Convergence in terms of total potential energy in case of elastic problem.

8 Beam elements

Ritz method is illustrated for a one-dimensional mechanical problem, the beam element, in the current chapter. Ritz method corresponds in this context an energy method, minimisation of the total potential energy of the system. Two variants based on slightly different theories, one for the classical beam theory and one for less slender beams, Timoshenko beam theory, are formulated. Selective reduced integration, chapter 7, will be demonstrated on the latter element.

8.1 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 12.1.

The solution of beam problems should make the total potential energy, see Eq. (12.10), minimum. Thus we want to find the minimum of this expression. This is the Ritz method, see section 4.5. It is a special case of the Galerkin method. The approximate solution should fulfil essential boundary conditions and minimise the energy expression

$$\Pi = \int_L \frac{1}{2} EI \left(\frac{d^2 w}{dx^2} \right)^2 dx - \int_L qwdx \tag{8.1}$$

Fulfilling essential boundary conditions are done by assigning appropriate nodal values to the model as will be seen in section 8.3. The element stiffness matrix and load vector will be formulated by applying Eq. (8.1) to one element. A few notes are at place first.

The derivatives in the functional are higher than in the example in section 6.2. The fundamental equilibrium equation expressed in displacement, Eq. (12.9), is a fourth order differential equation. The approximate solution must at least be able to represent constant curvature $\left(\kappa = -\frac{d^2 w}{dx^2} \right)$. Thus it needs to be at least cubic. The function must give continuous derivatives/slope between elements. Otherwise there is a risk that the split of the integral Eq. (8.1) 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-continuity*[17] 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. Then a two node beam element will have two degree of freedoms per element, see Figure 8.1. They can define the four coefficients of a third order polynomial. The element field is written as

$$w(x) = \boldsymbol{Nu} = \begin{bmatrix} N_{w1} & N_{\theta 1} & N_{w2} & N_{\theta 2} \end{bmatrix} \boldsymbol{u} \tag{8.2}$$

The shape functions are the polynomials shown below.

$$w(x) = \begin{bmatrix} 1 - 3\frac{x^2}{l^{e^2}} + 2\frac{x^3}{l^{e^3}} & -x + 2\frac{x^2}{l^e} - \frac{x^3}{l^{e^2}} & 3\frac{x^2}{l^{e^2}} - 2\frac{x^3}{l^{e^3}} & \frac{x^2}{l^e} - \frac{x^3}{l^{e^2}} \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} \tag{8.3}$$

These functions have the properties

$$\begin{aligned} &N_{w1}(0) = 1 \ , N_{w1}(l^e) = 0 \\ &N_{\theta 1}(0) = 0 \ , N_{\theta 1}(l^e) = 0 \\ &N_{w2}(0) = 0 \ , N_{w2}(l^e) = 1 \\ &N_{\theta 2}(0) = 0 \ , N_{\theta 2}(l^e) = 0 \end{aligned} \tag{8.4}$$

and

$$\begin{aligned} &\frac{dN_{w1}}{dx}(0) = 0 \ , \frac{dN_{w1}}{dx}(l^e) = 0 \\ &\frac{dN_{\theta 1}}{dx}(0) = -1 \ , \frac{dN_{\theta 1}}{dx}(l^e) = 0 \\ &\frac{dN_{w2}}{dx}(0) = 0 \ , \frac{dN_{w2}}{dx}(l^e) = 0 \\ &\frac{dN_{\theta 2}}{dx}(0) = 0 \ , \frac{dN_{\theta 2}}{dx}(l^e) = -1 \end{aligned} \tag{8.5}$$

The cubic equation is the exact solution to a beam bending problem where only nodal loads are applied, see Appendix 12.1.

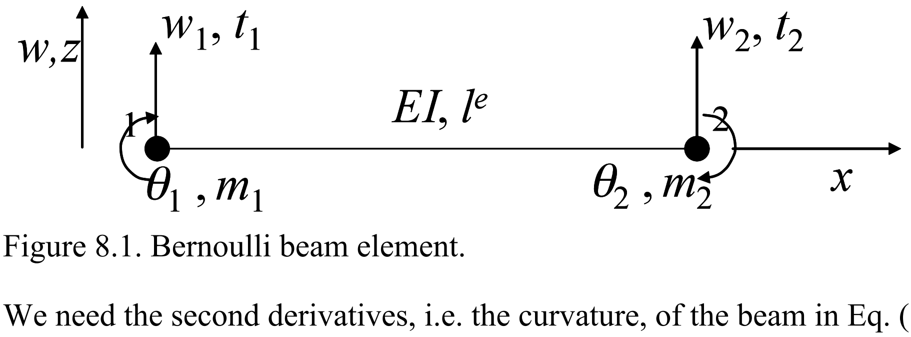


Figure 8.1. Bernoulli beam element.

We need the second derivatives, i.e. the curvature, of the beam in Eq. (8.1). This is

$$\kappa(x) = -\frac{d^2\boldsymbol{N}}{dx^2}\boldsymbol{u} = \boldsymbol{Bu} \tag{8.6}$$

with.

[17] 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.

$$\boldsymbol{B}=\left[\frac{6}{l^{e^2}}-12\frac{x}{l^{e^3}} \quad -\frac{4}{l^e}+6\frac{x}{l^{e^2}} \quad -\frac{6}{l^{e^2}}+12\frac{x}{l^{e^3}} \quad -\frac{2}{l^e}+6\frac{x}{l^{e^2}}\right] \tag{8.7}$$

Then Eq. (8.1) is written as, assuming only one element in the model,

$$\Pi=\int_{l^e}\frac{1}{2}EI(\boldsymbol{Bu})^T\boldsymbol{Bu}dx-\int_{l^e}(\boldsymbol{Nu})^T qdx \tag{8.8}$$

$$\Pi=\frac{1}{2}\boldsymbol{u}^T\int_{l^e}EI\boldsymbol{B}^T\boldsymbol{B}dx\boldsymbol{u}-\boldsymbol{u}^T\int_{l^e}\boldsymbol{N}^T qdx \tag{8.9}$$

$$\Pi=\frac{1}{2}\boldsymbol{u}^T\boldsymbol{ku}-\boldsymbol{u}^T\boldsymbol{f}_{ext} \tag{8.10}$$

Now we can identify element stiffness matrix as

$$\boldsymbol{k}=\int_{l^e}EI\boldsymbol{B}^T\boldsymbol{B}dx \tag{8.11}$$

and <u>consistent</u> nodal load vector due to the distributed load q

$$\boldsymbol{f}_{ext}=\int_{l^e}\boldsymbol{N}^T qdx \tag{8.12}$$

The word consistent means that the nodal forces and moments in $\boldsymbol{f}_{ext}$ give the same contribution to the potential of the loads as the original distributed load would have given in Eq. (8.1).

The stiffness matrix becomes

$$\boxed{\boldsymbol{k}=\int_{l^e}EI\begin{bmatrix}\frac{6}{l^{e^2}}-\frac{12x}{l^{e^3}}\\ -\frac{4}{l^e}+\frac{6x}{l^{e^2}}\\ -\frac{6}{l^{e^2}}+\frac{12x}{l^{e^3}}\\ -\frac{2}{l^e}+\frac{6x}{l^{e^2}}\end{bmatrix}\left[\frac{6}{l^{e^2}}-\frac{12x}{l^{e^3}} \quad -\frac{4}{l^e}+\frac{6x}{l^{e^2}} \quad -\frac{6}{l^{e^2}}+\frac{12x}{l^{e^3}} \quad -\frac{2}{l^e}+\frac{6x}{l^{e^2}}\right]dx \qquad (8.13)}$$

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

$$\boldsymbol{k}=\frac{EI}{L^3}\begin{bmatrix}12 & -6L & -12 & -6L\\ -6L & 4L^2 & 6L & 2L^2\\ -12 & 6L & 12 & 6L\\ -6L & 2L^2 & 6L & 4L^2\end{bmatrix} \tag{8.14}$$

and for the load, assuming q=constant along the beam

$$\mathbf{f}_{ext}=\begin{bmatrix}t_1\\ m_1\\ t_2\\ m_2\end{bmatrix}=\int_{l^e}\begin{bmatrix}1-3\frac{x^2}{l^{e^2}}+2\frac{x^3}{l^{e^3}}\\ -x+2\frac{x^2}{l^e}-\frac{x^3}{l^{e^2}}\\ 3\frac{x^2}{l^{e^2}}-2\frac{x^3}{l^{e^3}}\\ \frac{x^2}{l^e}-\frac{x^3}{l^{e^2}}\end{bmatrix}qdx=\begin{bmatrix}\frac{1}{2}\\ -\frac{l^e}{12}\\ \frac{1}{2}\\ \frac{l^e}{12}\end{bmatrix}ql^e \tag{8.15}$$

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.

An analysis of a beam problem is shown in section 8.3.

8.2 Timoshenko beam

The Timoshenko beam element in Figure 8.1 is formulated directly by use of the Tonti diagram in Figure 8.2. See appendix, section 12.2, for more about the Timoshenko beam theory.

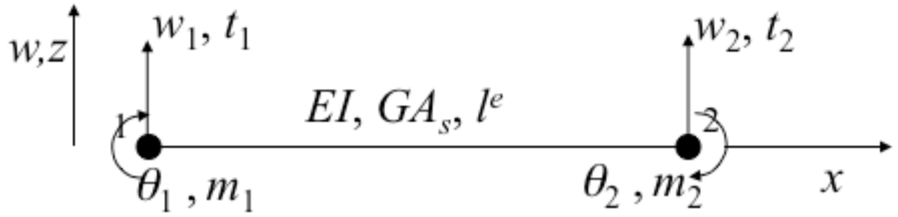


Figure 8.2. Tonti diagram for a two node Timoshenko beam element.

We use an isoparametric approach by

$$x(s) = N_1 x_1 + N_2 x_2 \tag{8.16}$$

with

$$N_1 = \frac{1}{2}(1-s) \quad N_2 = \frac{1}{2}(1+s) \tag{8.17}$$

The rotation and displacements are interpolated independently as

$$\mathbf{u}(s) = \begin{bmatrix} w \\ \theta \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{Nu} \tag{8.18}$$

This is a large difference to the Bernoulli beam case in Eq. (8.3). This gives

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{d\theta}{dx} \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 & \frac{dN_1}{dx} & 0 & \frac{dN_2}{dx} \\ \frac{dN_1}{dx} & N_1 & \frac{dN_2}{dx} & N_2 \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{Bu} \tag{8.19}$$

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 (6.34). The Jacobian of the mapping is

$$J \frac{dx}{ds} = \frac{l^e}{2} \tag{8.20}$$

where l^e is the length of the element. The **B**-matrix becomes then

$$\mathbf{B} = \begin{bmatrix} 0 & -\frac{1}{l^e} & 0 & \frac{1}{l^e} \\ -\frac{1}{l^e} & \frac{1}{2}(1-s) & \frac{1}{l^e} & \frac{1}{2}(1+s) \end{bmatrix} \tag{8.21}$$

The total potential energy[18], section 12.2, is

$$\Pi = \int_L \frac{1}{2}\begin{bmatrix}\dfrac{d\theta}{dx} & \gamma\end{bmatrix}\begin{bmatrix}EI & 0\\ 0 & \dfrac{GA}{\alpha}\end{bmatrix}\begin{bmatrix}\dfrac{d\theta}{dx}\\ \gamma\end{bmatrix}dx - \int_L qwdx \tag{8.22}$$

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

$$\mathbf{k} = \int_{l^e}\mathbf{B}^T\mathbf{E}\mathbf{B}dx \tag{8.23}$$

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

$$\mathbf{k} = \int_{-1}^{1}\begin{bmatrix}\dfrac{GA_s}{l^{e2}} & \dfrac{-GA_s}{2l^e}(1-s) & \dfrac{-GA_s}{l^{e2}} & \dfrac{-GA_s}{2l^e}(1+s)\\ \dfrac{-GA_s}{2l^e}(1-s) & \dfrac{EI}{l^{e2}}+\dfrac{GA_s}{4}(1-s)^2 & \dfrac{GA_s}{2l^e}(1-s) & \dfrac{-EI}{l^{e2}}+\dfrac{GA_s}{4}(1-s^2)\\ \dfrac{-GA_s}{l^{e2}} & \dfrac{GA_s}{2l^e}(1-s) & \dfrac{GA_s}{l^{e2}} & \dfrac{GA_s}{2l^e}(1+s)\\ \dfrac{-GA_s}{2l^e}(1+s) & \dfrac{-EI}{l^{e2}}+\dfrac{GA_s}{4}(1-s^2) & \dfrac{GA_s}{2l^e}(1+s) & \dfrac{EI}{l^{e2}}+\dfrac{GA_s}{4}(1+s)^2\end{bmatrix}\frac{l^e}{2}ds$$

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. (8.10) and can therefore also be uncoupled[18]. It is written as

$$\mathbf{k} = \mathbf{k}_b + \mathbf{k}_s \tag{8.24}$$

where

$$\mathbf{k}_b = \int_{-1}^{1}\frac{EI}{l^{e2}}\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 1 & 0 & -1\\ 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 1\end{bmatrix}\frac{l^e}{2}ds = \frac{EI}{l^e}\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 1 & 0 & -1\\ 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 1\end{bmatrix} \tag{8.25}$$

and

[18] 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.

$$\mathbf{k}_s = \int_{-1}^{1} \frac{GA_s}{l^{e2}} \begin{bmatrix} 1 & -\frac{l^e}{2}(1-s) & -1 & -\frac{l^e}{2}(1+s) \\ -\frac{l^e}{2}(1-s) & \left(\frac{l^e}{2}\right)^2 (1-s)^2 & \frac{l^e}{2}(1-s) & \left(\frac{l^e}{2}\right)^2 (1-s^2) \\ -1 & \frac{l^e}{2}(1-s) & 1 & \frac{l^e}{2}(1+s) \\ -\frac{l^e}{2}(1+s) & \left(\frac{l^e}{2}\right)^2 (1-s^2) & \frac{l^e}{2}(1+s) & \left(\frac{l^e}{2}\right)^2 (1+s)^2 \end{bmatrix} \frac{l^e}{2} ds \tag{8.26}$$

Exact integration gives for the latter

$$\mathbf{k}_s = \frac{GA_s}{l^e} \begin{bmatrix} 1 & -l^e/2 & -1 & -l^e/2 \\ -l^e/2 & l^{e2}/3 & l^e/2 & l^{e2}/6 \\ -1 & l^e/2 & 1 & l^e/2 \\ -l^e/2 & l^{e2}/6 & l^e/2 & l^{e2}/3 \end{bmatrix} \tag{8.27}$$

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. (8.26) are underintegrated. We use one integration point formula according to Table 7.1 and get

$$\mathbf{k}_s(2,2) = \frac{GA_s l^e}{8 l^e} \int_{-1}^{1} (1-s)^2 ds = \frac{GA_s l^e}{8} (1-0)^2 \cdot 2 = \frac{GA_s l^e}{4} \tag{8.28}$$

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

$$\mathbf{k}_s(2,4) = \mathbf{k}_s(4,2) = \frac{GA_s l^e}{8 l^e} \int_{-1}^{1} (1-s^2) ds = \frac{GA_s l^e}{8} (1-0)^2 \cdot 2 = \frac{GA_s l^e}{4} \tag{8.29}$$

Then we have

$$\tilde{\mathbf{k}}_s = \frac{GA_s}{l^e} \begin{bmatrix} 1 & -l^e/2 & -1 & -l^e/2 \\ -l^e/2 & l^{e2}/4 & l^e/2 & l^{e2}/4 \\ -1 & l^e/2 & 1 & l^e/2 \\ -l^e/2 & l^{e2}/4 & l^e/2 & l^{e2}/4 \end{bmatrix} \tag{8.30}$$

The <u>consistent</u> nodal load vector due to the distributed load ***q*** is

$$\mathbf{f}_{ext} = \int_{-1}^{1} \frac{1}{2} \begin{bmatrix} (1-s) & 0 \\ 0 & (1-s) \\ (1+s) & 0 \\ 0 & (1+s) \end{bmatrix} \begin{bmatrix} q \\ m \end{bmatrix} \frac{l^e}{2} ds \tag{8.31}$$

Notice the extension of ***q*** to a vector with two parts. One is related to the deflection of the beam and the other to its rotation. Originally, Eq. (8.31) comes from the potential of the loads that is the loads multiplying the deformations in Eq. (8.22). Then we had only *qw* in the product. Now we have, see Eq. (8.18), 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. (8.31). Integration of above for constant distributed loads gives simply

$$\mathbf{f}_{ext} = \begin{bmatrix} \frac{ql^e}{2} \\ \frac{ml^e}{2} \\ \frac{ql^e}{2} \\ \frac{ml^e}{2} \end{bmatrix} \qquad (8.32)$$

8.3 Cantilever beam problem

We will compare finite element solutions with the analytic solution according to Bernoulli beam theory in section 12.1. A model with one element will be used to solve the cantilever beam with a point load, upper part in Figure 8.3. The reaction force and moment are replacing the wall that was drawn in Figure 12.5. They are unknown and their magnitudes comes from the fixed displacement and rotation at the wall, *x*=0.

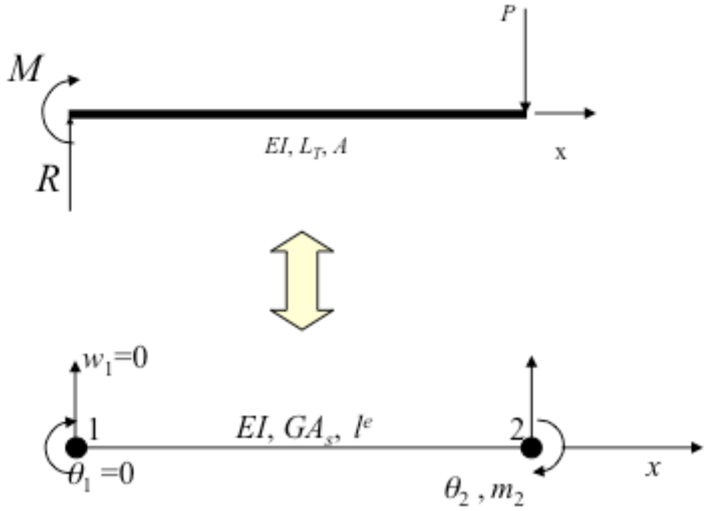


Figure 8.3. Cantilever beam with point load and a one element model.

The problem is shown and solved analytically in section 12.1. The solution is

The solution is thus

$$w(x) = -\frac{PL}{2EI}x^2 + \frac{P}{6EI}x^3 = \frac{PL^3}{EI}\left(\frac{1}{6}\left(\frac{x}{L}\right)^3 - \frac{1}{2}\left(\frac{x}{L}\right)^2\right) \qquad (8.33)$$

The maximum displacement (downwards) is

$$w_{BBT} = -\frac{PL^3}{3EI}$$

The one element model is shown in the lower part of Figure 8.3. 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. The in 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

$$\frac{EI}{L^3}\begin{bmatrix}12 & 6L\\ 6L & 4L^2\end{bmatrix}\begin{bmatrix}w_2\\\theta_2\end{bmatrix}=\begin{bmatrix}-P\\0\end{bmatrix} \tag{8.34}$$

giving

$$\begin{bmatrix}w_2\\\theta_2\end{bmatrix}=\frac{L}{12EI}\begin{bmatrix}4L^2 & -6L\\ -6L & 12\end{bmatrix}\begin{bmatrix}-P\\0\end{bmatrix}=\frac{PL^3}{EI}\begin{bmatrix}-\frac{1}{3}\\ \frac{1}{2}\end{bmatrix} \tag{8.35}$$

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.

$$w_{LBB}=-\frac{PL^3}{3EI^3}\left[1+\left(\frac{H}{L}\right)^2\frac{(12+11\nu)}{20}\right] \tag{12.32}$$

We get

$$w_2=-\frac{PL^3}{3EI}=w_{BBT} \tag{8.36}$$

The fully integrated Timoshenko beam, Eq. (8.25) and Eq. (8.27), gives,

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

This leads to

$$\begin{bmatrix}\frac{GA_s}{L} & \frac{GA_s}{2}\\ \frac{GA_s}{2} & \frac{EI}{L}+\frac{GA_sL}{3}\end{bmatrix}\begin{bmatrix}w_2\\\theta_2\end{bmatrix}=\begin{bmatrix}-P\\0\end{bmatrix} \tag{8.37}$$

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

$$f_s=GA_sL\frac{L}{EI} \tag{8.38}$$

Eq. (8.34) can be written as

$$\frac{EI}{L}\begin{bmatrix}\frac{f_s}{L^2} & \frac{f_s}{2L}\\ \frac{f_s}{2L} & 1+\frac{f_s}{3}\end{bmatrix}\begin{bmatrix}w_2\\\theta_2\end{bmatrix}=\begin{bmatrix}-P\\0\end{bmatrix} \tag{8.39}$$

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

$$w_2 = -\frac{PL^3}{EI} \frac{\left(\frac{1}{f_s}+\frac{1}{3}\right)}{\left(\frac{1}{f_s}+\frac{1}{12}\right)} \tag{8.40}$$

Assuming a square cross-section of the beam gives

$$I = \frac{BH^3}{12}, A = BH, \ \alpha = \frac{5}{6}, G = \frac{E}{2(1+\nu)} \tag{8.41}$$

Insertion of above into Eq. (8.38) gives

$$f_s = \frac{1}{6(1+\nu)} \frac{12BHL^2}{BH^3\frac{5}{6}} = \frac{1}{2(1+\nu)}\left(\frac{L}{H}\right)^2 \approx \left(\frac{L}{H}\right)^2 \tag{8.42}$$

This makes the deflection of the edge of the beam, Eq.(8.40),

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

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

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

Then the computer will experience Eq. (8.39) as

$$\frac{EI}{L} \begin{bmatrix} \frac{f_s}{L^2} & \frac{f_s}{2L} \\ \frac{f_s}{2L} & \frac{f_s}{3} \end{bmatrix} \begin{bmatrix} w_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -P \\ 0 \end{bmatrix} \tag{8.43}$$

This gives the solution

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

Leading to

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

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. (8.30). That formulation gives

$$\frac{EI}{L}\begin{bmatrix}\frac{f_s}{L^2} & \frac{f_s}{2L}\\ \frac{f_s}{2L} & 1+\frac{f_s}{4}\end{bmatrix}\begin{bmatrix}w_2\\ \theta_2\end{bmatrix}=\begin{bmatrix}-P\\ 0\end{bmatrix} \tag{8.44}$$

The solution becomes

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

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

leading to

$$w_2=-\frac{PL^3}{EI}\left(2(1+\nu)\left(\frac{H}{L}\right)^2+\frac{1}{4}\right)\approx\left[\text{if }\frac{H}{L}<<1\right]=-\frac{PL^3}{4EI}=0.75w_L \tag{8.45}$$

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 8.1. Computed maximum deflection of cantilever beam. Finite element results are normalised versus theoretical result. 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

9 Isoparametric mapping in two dimensions

The concept of isoparametric mapping was introduced in section 6.3 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 chapter 10 where the isoparametric mapping is a necessary step in the finite element formulation. There it is shown for a two-dimensional element.

Chapter **Error! Reference source not found.** described the numerical integration technique. This gives added possibilities to tailor elements as was described in the case of the Timoshenko beam in section 8.2. 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 11.

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 chapter 10. 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 $dA=dxdy$ 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 9.1.

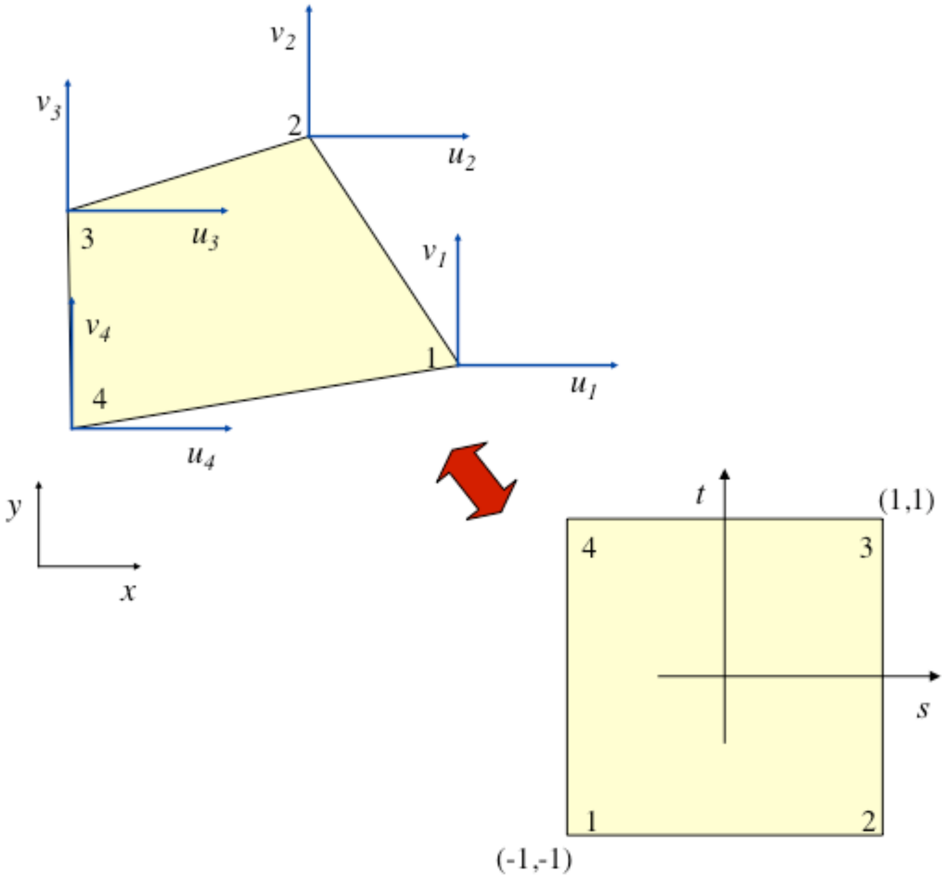


Figure 9.1. Isoparametric mapping of a four node element in two dimensions. The element is a square between (-1,-1) to (1,1) in the local coordinate system.

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

$$x(s,t)=\sum_{i=1}^{nnode} N(s,t)_i x_i$$
$$y(s,t)=\sum_{i=1}^{nnode} N(s,t)_i y_i \qquad (9.1)$$

It can be written in matrix form as

$$\boldsymbol{x}=\boldsymbol{Nc} \qquad (9.2)$$

The interpolation of the displacement fields is written in the same format, see Eq. (10.1) 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. (9.1), 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

$$\frac{\partial h(s(x,y),t(x,y))}{\partial x}=\frac{\partial h}{\partial s}\frac{\partial s}{\partial x}+\frac{\partial h}{\partial t}\frac{\partial t}{\partial x} \qquad (9.3)$$

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

$$\begin{bmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\end{bmatrix}=\begin{bmatrix}\frac{\partial s}{\partial x} & \frac{\partial t}{\partial x}\\ \frac{\partial s}{\partial y} & \frac{\partial t}{\partial y}\end{bmatrix}\begin{bmatrix}\frac{\partial}{\partial s}\\ \frac{\partial}{\partial t}\end{bmatrix}=\mathbf{J}^{-1}\begin{bmatrix}\frac{\partial}{\partial s}\\ \frac{\partial}{\partial t}\end{bmatrix} \tag{9.4}$$

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. (9.4)

$$\begin{bmatrix}\frac{\partial}{\partial s}\\ \frac{\partial}{\partial t}\end{bmatrix}=\begin{bmatrix}\frac{\partial x}{\partial s} & \frac{\partial y}{\partial s}\\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t}\end{bmatrix}\begin{bmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\end{bmatrix}=\boldsymbol{J}\begin{bmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\end{bmatrix} \tag{9.5}$$

The terms in the Jacobian matrix, **J**, can be computed from the isoparametric mapping in Eq. (9.1) as

$$\begin{aligned}J_{11}&=\frac{\partial x}{\partial s}=\frac{\partial}{\partial s}\left(\sum_{i=1}^{nnode}N(s,t)_i x_i\right)=\sum_{i=1}^{nnode}\frac{\partial N_i}{\partial s}x_i\ ,\ J_{21}=\frac{\partial x}{\partial t}=\sum_{i=1}^{nnode}\frac{\partial N_i}{\partial t}x_i\\ J_{12}&=\frac{\partial y}{\partial s}=\frac{\partial}{\partial s}\left(\sum_{i=1}^{nnode}N(s,t)_i y_i\right)=\sum_{i=1}^{nnode}\frac{\partial N_i}{\partial s}y_i\ ,\ J_{22}=\frac{\partial y}{\partial t}=\sum_{i=1}^{nnode}\frac{\partial N_i}{\partial t}y_i\end{aligned} \tag{9.6}$$

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

$$\begin{bmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\end{bmatrix}=[J]^{-1}\begin{bmatrix}\frac{\partial}{\partial s}\\ \frac{\partial}{\partial t}\end{bmatrix}=\frac{1}{\det\boldsymbol{J}}\begin{bmatrix}J_{22} & -J_{12}\\ -J_{21} & J_{11}\end{bmatrix}\begin{bmatrix}\frac{\partial}{\partial s}\\ \frac{\partial}{\partial t}\end{bmatrix} \tag{9.7}$$

where

$$|\boldsymbol{J}|=\det\boldsymbol{J}=J_{11}J_{22}-J_{12}J_{21}=\frac{\partial x}{\partial s}\frac{\partial y}{\partial t}-\frac{\partial x}{\partial s}\frac{\partial y}{\partial t} \tag{9.8}$$

The above derivations are shown in Figure 9.2 where the information in the derivatives and the determinant of the Jacobian are hopefully clearer. It shows how an infinitesimal area element *dA=dsdt* becomes a parallelogram. The mapping of a small area element is always a linear mapping.

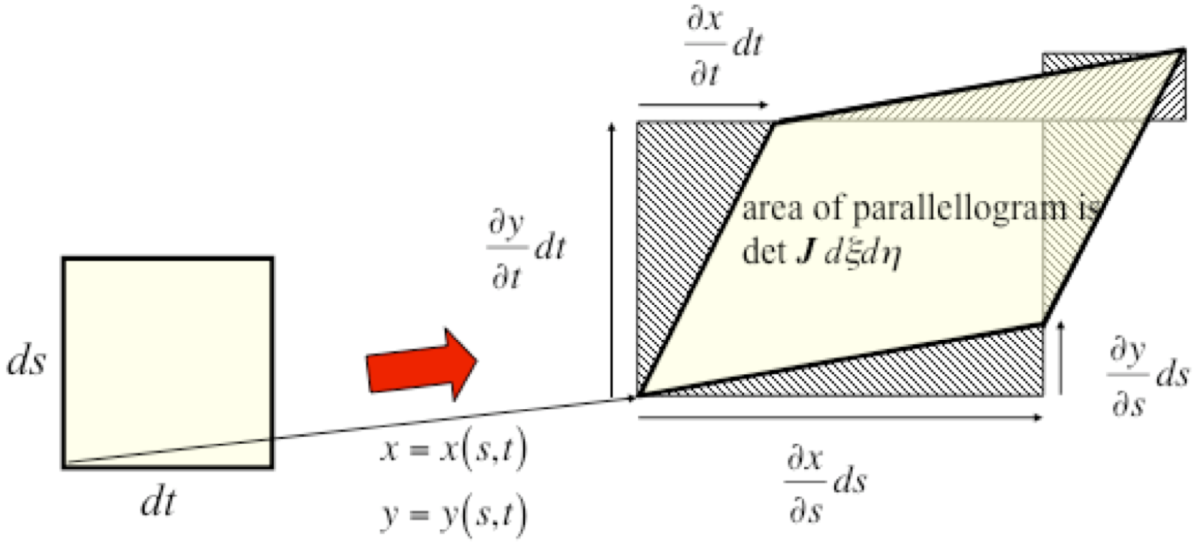

$\frac{\partial x}{\partial t}dt$
$\frac{\partial y}{\partial t}dt$
area of parallellogram is
det $\boldsymbol{J}\, d\xi d\eta$
ds
$\frac{\partial y}{\partial s}ds$
dt
$x = x(s,t)$
$y = y(s,t)$
$\frac{\partial x}{\partial s}ds$

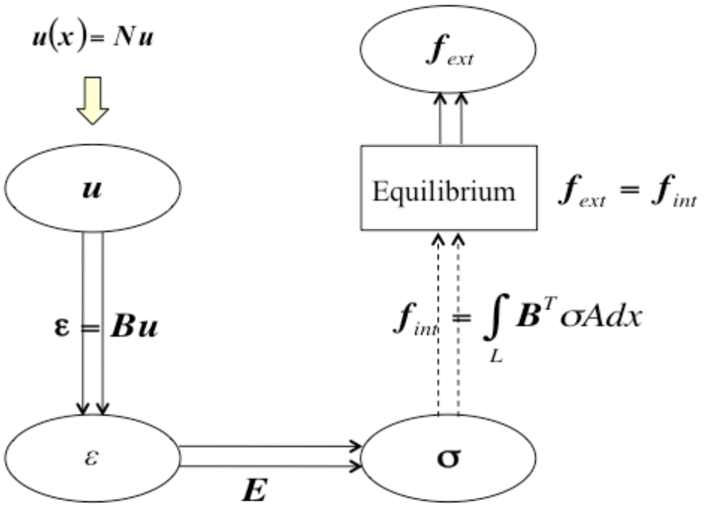
Figure 9.2. Change in scale due to change of variables.

10 A four node plane stress element

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 stress[19] problems will be derived. The basic continuum mechanics[20] 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 10.1

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

[20] http://en.wikipedia.org/wiki/Strain_(materials_science), http://en.wikipedia.org/wiki/Strain_tensor#Infinitesimal_strain_tensor, http://en.wikipedia.org/wiki/Hooke%27s_law



$$k = \int_{V^e} B^T E B dV \qquad f_{ext} = \int_V N^T f(x)_{ext} dV$$

Figure 10.1. The shape functions determines 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

$$u(x) = N(x)u \tag{10.1}$$

where the shape functions in the matrix N fulfils some requirements listed below.

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

$$N_J(\mathbf{x}_K) = \begin{cases} 1 \text{ if } J = K \\ 0 \text{ if } J \neq K \end{cases} \tag{10.2}$$

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

$$\sum_J N_J(\mathbf{x}) = 1 \tag{10.3}$$

3. The shape functions must be sufficient continuous. If the fundamental equation to be solved has an 2*m*:th derivative, then the WRM formulation after appropriate partial integration[21] 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

$$\varepsilon(x) = B(x)u \tag{10.4}$$

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

[21] 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.

$$\boldsymbol{\sigma} = \boldsymbol{E\varepsilon} \tag{10.5}$$

Combining these, see Figure 10.1, leads to a general formula for the stiffness matrix

$$\boldsymbol{k} = \int_{V^e} \boldsymbol{B}^T \boldsymbol{EB} dV \tag{10.6}$$

and the consistent load vector

$$\boldsymbol{f}_{ext} = \int_{V} \boldsymbol{N}^T \boldsymbol{f}(x)_{ext} dV \tag{10.7}$$

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

10.1 Displacements and geometry of four-node plane element

The local coordinate system of the element is shown in Figure 10.2. This is the basis for formulating the element and where the element integrals, Eq. (10.6) and Eq. (10.7) are solved. The shape functions are

$$\left.\begin{aligned} N_1 &= \frac{1}{4}(1-s)(1-t) \\ N_2 &= \frac{1}{4}(1+s)(1-t) \\ N_3 &= \frac{1}{4}(1+s)(1+t) \\ N_4 &= \frac{1}{4}(1-s)(1+t) \end{aligned}\right\} \Leftrightarrow N_i = \frac{1}{4}(1+s_i s)(1+t_i t) \tag{10.8}$$

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

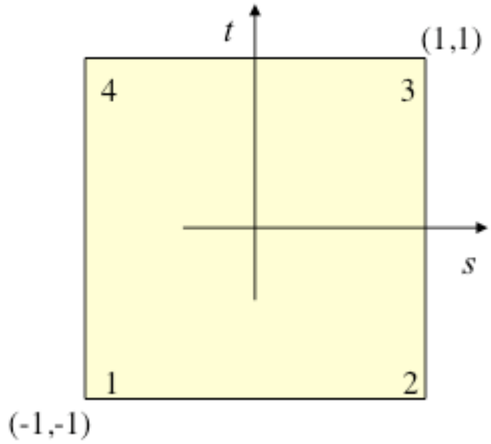


Figure 10.2. Local coordinate system of plane, four node element.

The isoparametric mapping from the local to the global coordinate system, see Figure 9.1, can be written in some variants

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

$$\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & v \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \mathbf{Nc} \tag{10.9}$$

The first variant with a sum over *nnode* 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. **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

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

$$\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & v \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \mathbf{Nu} \tag{10.10}$$

10.2 Strains of four-node plane element

The strain components needed for plane stress are

$$\varepsilon = \begin{bmatrix} \varepsilon_x(x,y) \\ \varepsilon_y(x,y) \\ \gamma_{xy}(x,y) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{bmatrix} \tag{10.11}$$

Insertion of the shape functions, Eq. (10.10), for the four node element gives

$$\varepsilon = \begin{bmatrix} \varepsilon_x(s,t) \\ \varepsilon_y(s,t) \\ \gamma_{xy}(s,t) \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix} \left\{ \begin{matrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{matrix} \right\} \tag{10.12}$$

It can be written in the node summation style as

$$\varepsilon = \sum_{i=1}^{nnode} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \sum_{i=1}^{nnode} \mathbf{B}_i \mathbf{u}_i = \mathbf{Bu} \tag{10.13}$$

The determination of global derivative of the shape functions are done as shown in chapter 9. The local derivatives of the shape functions are set up first. The Jacobian is determined by Eq. (9.6). Notice that it depends on the coordinates of the element as expected. Thereafter Eq. (9.7) is used to calculate the global derivatives needed for Eq. (10.12). The strains are explicitly expressed in the local coordinates. If the strain in needed ad a coordinate x, then the local coordinate must be solved first from Eq. (10.9) and thereafter the found local coordinate is inserted into Eq. (10.12). Eq. (10.9) can in the general case not be solved analytically but must be solved numerically.

10.3 Constitutive model for plane stress

The relation between strains and stresses, Eq. (10.5), is

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1+v} \begin{bmatrix} 1 & -v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \tag{10.14}$$

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

10.4 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 the solving the element integrals is given in Box 10.1. It is assumed that the material properties are constant over the element. The **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. (10.8), gives one linear term. Thus most of the terms in the integral for the stiffness matrix, Eq. (10.6), are constant or linear. However, the highest term is quadratic. This requires *nint* =2*2 integration points according to Eq. (7.2).

Box 10.1. Procedure for the computation of element stiffness matrix.

Set up constitutive matrix ***E***, Eq. (10.14).

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}[22]

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

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

Compute the Jacobian matrix, Eq. (9.6), and its determinant, Eq. (9.8), at the integration point

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

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

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

Compute the global derivatives of the shape functions according to Eq. (9.4)

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

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

11 Convective heat transfer

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 equation[23] in absence of heat sinks is written as

$$\rho c\dot{T} - \lambda \frac{d^2T}{d\chi^2} = 0 \qquad (11.1)$$

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

[23] http://en.wikipedia.org/wiki/Heat_conduction

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

$$x = \chi + vt \tag{11.2}$$

where v is the velocity. This transforms[24] Eq. (11.1) to

$$\rho c\left(\dot{T} + v\frac{dT}{dx}\right) - \lambda\frac{d^2T}{dx^2} = 0 \tag{11.3}$$

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

$$\rho cv\frac{dT}{dx} - \lambda\frac{d^2T}{dx^2} = 0 \tag{11.4}$$

The analytic solution of this problem with the boundary conditions

$$T(0) = T_{left} \tag{11.5}$$

$$T(L) = T_{right} \tag{11.6}$$

is

$$T(x) = T_{left} + \left(T_{right} - T_{left}\right)\frac{\left(e^{\frac{\rho c}{\lambda}x} - 1\right)}{\left(e^{\frac{\rho c}{\lambda}L} - 1\right)} \tag{11.7}$$

This 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

$$Pe = \frac{\rho cvl^e}{2\lambda} \tag{11.8}$$

The analytic solution can now be written as

$$T(x) = T_{left} + \left(T_{right} - T_{left}\right)\frac{\left(e^{2Pe\frac{x}{l^e}} - 1\right)}{\left(e^{2Pe\frac{L}{l^e}} - 1\right)} \tag{11.9}$$

The finite element solution leads to a global system of equations

$$\mathbf{KT} = \mathbf{0} \tag{11.10}$$

The standard contribution to **K** from the for conductivity matrix using a two-node linear element is, see section 12.3,

$$\mathbf{k}_{cond} = \int_{v^{(e)}} \mathbf{B}_{th}^T \lambda \mathbf{B}_{th} dv = \frac{\lambda}{l^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{11.11}$$

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. (8.1). Using the latter leads to an expression for consistent nodal loads shown in Eq. (8.12). Thus q is repleced by $\rho cv\frac{dT}{dx}$ in this equation. This gives

[24] http://en.wikipedia.org/wiki/Total_derivative

$$\int_{-1}^{1} \mathbf{N}^T \rho c v \frac{dT}{dx} J ds \tag{11.12}$$

The gradient term is

$$\mathbf{BT}^e = \frac{1}{l^e}\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} T_1 \\ T_2 \end{bmatrix}^e \tag{11.13}$$

Combining the last two equations gives

$$\int_{-1}^{1} \rho c v \mathbf{N}^T \mathbf{B} J ds \mathbf{T}^e = \mathbf{k}_{conv}\mathbf{T}^e = \frac{\rho c v}{2}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}\mathbf{T}^e \tag{11.14}$$

Thus it will also contribute to the **K**-matrix. Summing the two matrices, Eq. (11.13) and Eq. (11.14), and using Eq. (11.8) gives the combined element matrix

$$\mathbf{k} = \frac{\lambda}{l^e}\left[\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + Pe\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}\right] = \frac{\lambda}{l^e}\begin{bmatrix} 1-Pe & -1+Pe \\ -1-Pe & 1+Pe \end{bmatrix} \tag{11.15}$$

to be assembled into Eq (11.10). 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 ${}_{\mathrm{mod}}Pe$ according to the equation below.

$${}_{\mathrm{mod}}Pe = \tanh(Pe) \tag{11.16}$$

The classical formulation is compared with the stabilised formulation and theoretical solutions for *Pe*=1 in Figure 11.1 and *Pe*=3 in Figure 11.2.

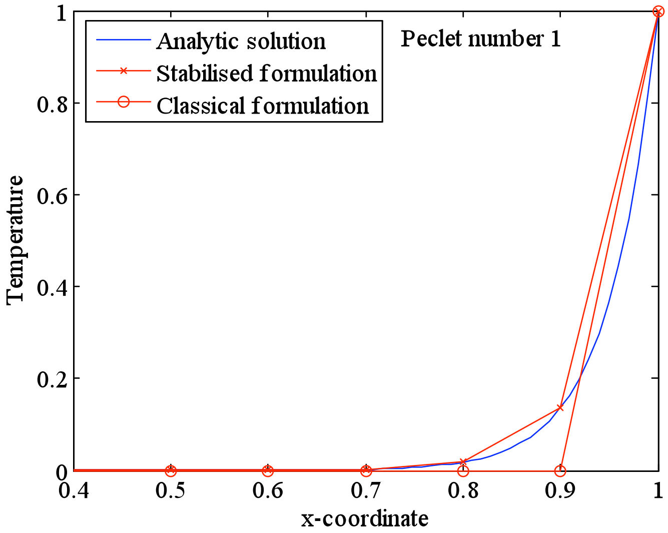


Figure 11.1. Temperature in fluid flow with given inlet and outlet temperatures for *Pe*=1.

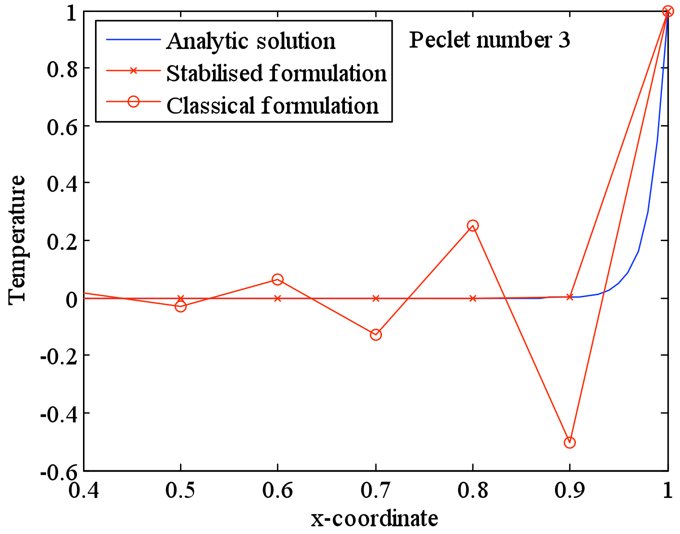


Figure 11.2. Temperature in fluid flow with given inlet and outlet temperatures for *Pe*=3.

The heat transfer in a material moving with the velocity *v* in positive direction and a point source at the coordinate x_c is

$$\rho c v \frac{dT}{dx} - \lambda \frac{d^2T}{dx^2} = P\delta\left(x - x_c\right) \tag{11.17}$$

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. (11.10) becomes

$$\boldsymbol{K}_{th}\boldsymbol{T} = \dot{\boldsymbol{Q}}_{ext} \tag{11.18}$$

The enhancement of the heat conductivity that gives ${}_{\text{mod}}Pe$ in Eq. (11.16) is

$${}_{\text{mod}}\lambda = \lambda\left(1 + \alpha Pe\right) \tag{11.19}$$

where

$$\alpha = \coth\left(Pe\right) - \frac{1}{Pe} \tag{11.20}$$

The value $P/\left(1 + \alpha Pe\right)$ is placed at the appropriate position in $\dot{\boldsymbol{Q}}_{ext}$ corresponding to the node at x_c. The classical formulation is compared with the stabilised formulation and theoretical solutions for *Pe*=1 in Figure 11.3 and *Pe*=3 in Figure 11.4.

The improved formulation using an enhanced convection is a prelude to Stream Upwind Petrov-Galerking (SUPG) formulations. This it is can be derived using a Petrov-Galerkin approach.

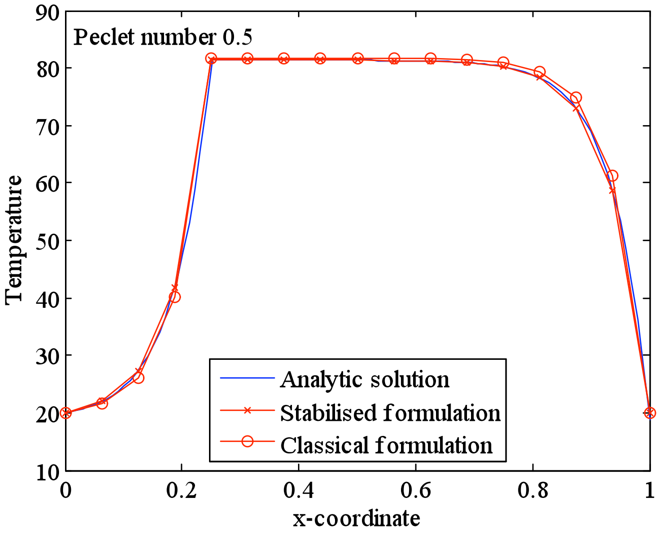


Figure 11.3. Temperature due to a point heat source moving to the left, *Pe*=0.5.

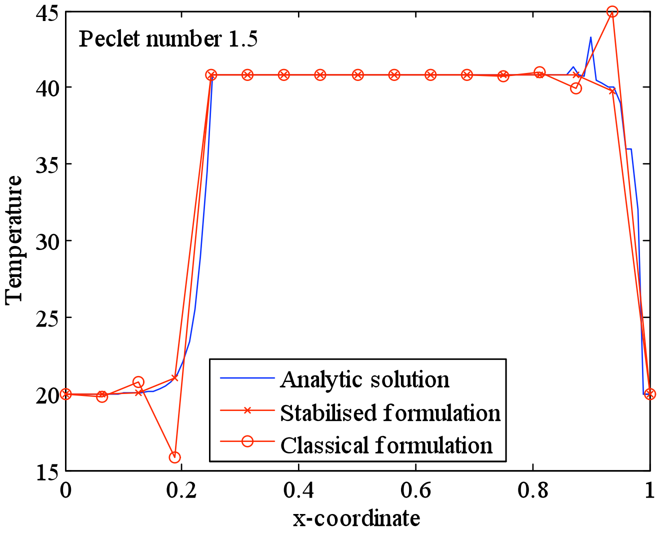


Figure 11.4. Temperature due to a point heat source moving to the left, *Pe*=1.5. Wiggles at end of analytic solution due to inaccuracy in its evaluation.

12 Appendix : Beam theories

12.1 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 12.1. Then the axial displacement due to a deflection of the centre line is

$$u = -z(-\theta) = z\theta = -z\frac{dw}{dx} \tag{12.1}$$

This gives a linear variation of the axial strain as

$$\varepsilon = \frac{du}{dx} = z\frac{d\theta}{dx} = -z\frac{d^2w}{dx^2} = z\kappa \tag{12.2}$$

All other strains are zero due to the assumed deformation in Eq. ((12.1)**.** The bending moment corresponding to the linear axial stress distribution, see Figure 12.2, becomes

$$M = \int_A \sigma z dA = E\int_A \varepsilon z dA = -E\frac{d^2w}{dx^2}\int_A z^2 dA \tag{12.3}$$

The last integral is the definition of area moment of inertia giving

$$M = -EI\frac{d^2w}{dx^2} = EI\kappa \tag{12.4}$$

The curvature has been introduced above as

$$\kappa = -\frac{d^2w}{dx^2} \tag{12.5}$$

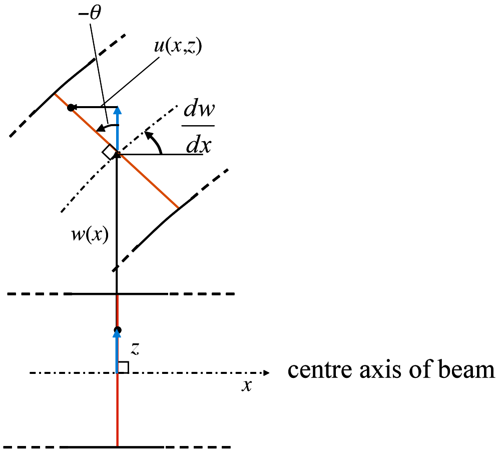


Figure 12.1. Assumed deformation of beam cross-section according to Bernoulli beam theory.

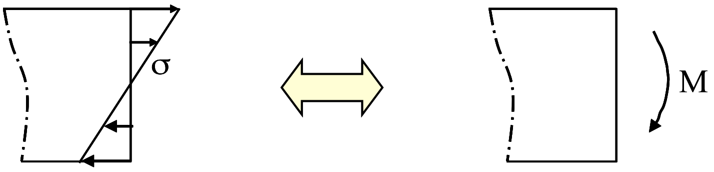


Figure 12.2. Axial stress distribution and equivalent bending moment.

Moment equilibrium for an infinitesimal element, Figure 12.3, 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

$$\frac{dM}{dx}=T \qquad (12.6)$$

Transverse force equilibrium gives

$$-T+\left(T+\frac{dT}{dx}dx\right)+qdx=0$$

leading to

$$\frac{dT}{dx}=-q \qquad (12.7)$$

Combining Eq. (12.6) and Eq. (12.7) gives

$$\frac{d^2M}{dx^2}=-q \qquad (12.8)$$

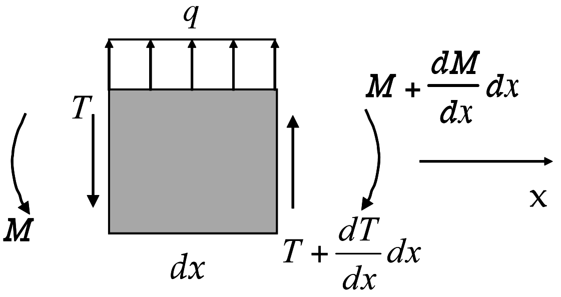


Figure 12.3. Equilibrium of infinitesimal piece of beam.

Insertion of Eq.(12.4) gives

$$\frac{d^4w}{dx^4}=\frac{q}{EI} \qquad (12.9)$$

This is the equilibrium equation for a beam expressed in the displacement field $w(x)$. The basic relations are summarised in Figure 12.4. 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 (12.4) and (12.6).

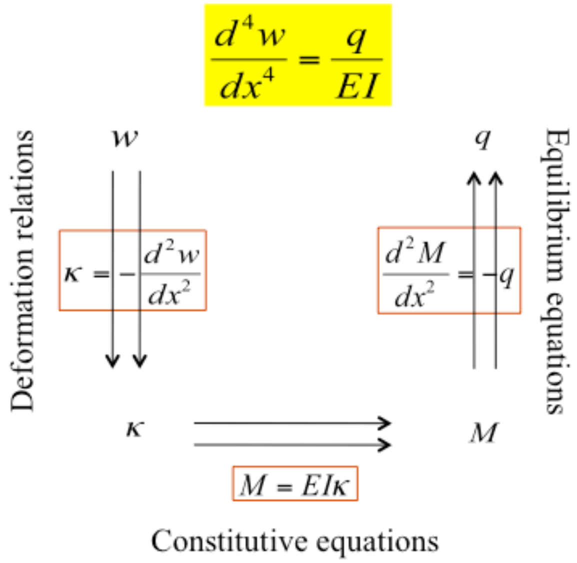


Figure 12.4. Tonti diagram for beam relations

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

$$\Pi=\int_L\int_A\frac{1}{2}\sigma\varepsilon\ dAdx-\int_L qwdx=\int_L\int_A\frac{1}{2}E\varepsilon^2dAdx-\int_L qwdx$$

Hooke's law, $\sigma=E\varepsilon$, 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

$$\Pi=\int_L\int_A\frac{1}{2}E\left(-z\frac{d^2w}{dx^2}\right)^2dAdx-\int_L qwdx$$

Some quantities does only vary with the x-coordinate

$$\Pi=\int_L\frac{1}{2}E\left(\frac{d^2w}{dx^2}\right)^2\int_A z^2dAdx-\int_L qwdx$$

Use of the definition of area moment of inertia gives finally

$$\Pi=\int_L\frac{1}{2}EI\left(\frac{d^2w}{dx^2}\right)^2dx-\int_L qwdx \qquad (12.10)$$

A cantilever beam problem with a point load at the end as shown in Figure 12.5 is defined by Eq. (12.9) and the appropriate boundary conditions;

no displacement near wall

$$w(0)=0 \qquad (12.11)$$

and no slope near wall

$$\left.\frac{dw}{dx}\right|_{x=0}=0 \qquad (12.12)$$

No moment at right edge

$$\left.\frac{d^2w}{dx^2}\right|_{x=L}=0 \qquad (12.13)$$

Given transverse force at right edge

$$T = -EI\frac{d^3w}{dx^3}\bigg|_{x=L} = P \tag{12.14}$$

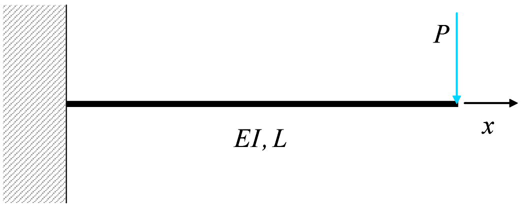


Figure 12.5. Cantilever beam with point load.

Integration of Eq. (12.9) four times with q=0 gives

$$w = A + Bx + Cx^2 + Dx^3 \tag{12.15}$$

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 8.1. The two first boundary condition at x=0 gives A=B=0.

The boundary condition Eq. (12.14) gives

$$D = \frac{P}{6EI} \tag{12.16}$$

and the use of Eq. (12.13) leads to

$$2C + 6DL = 2C + \frac{PL}{EI} = 0 \rightarrow C = \frac{PL}{2EI} \tag{12.17}$$

The solution is thus

$$w(x) = -\frac{PL}{2EI}x^2 + \frac{P}{6EI}x^3 = \frac{PL^3}{EI}\left(\frac{1}{6}\left(\frac{x}{L}\right)^3 - \frac{1}{2}\left(\frac{x}{L}\right)^2\right) \tag{12.18}$$

The maximum displacement (downwards) is

$$w_{BBT} = -\frac{PL^3}{3EI} \tag{12.19}$$

12.2 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 12.1 but now as in Figure 12.6. 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

$$\tau = G\gamma = \frac{E}{2(1+\nu)}\gamma \tag{12.20}$$

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

$$u = -z(-\theta) = z\theta = -z\left(\frac{dw}{dx} - \gamma\right) \tag{12.21}$$

This gives a linear variation of the axial strain as

$$\varepsilon = \frac{du}{dx} = z\frac{d\theta}{dx} \qquad (12.22)$$

The axial strain cannot be related to the second derivative *w* anymore. Accordingly the formula for the cross-sectional moment, see Figure 12.2, becomes somewhat different than for the Bernoulli beam. It becomes

$$M = \int_A \sigma z dA = E\int_A \varepsilon z dA = E\left(\frac{d\theta}{dx}\right)\int_A z^2 d = EI\frac{d\theta}{dx} \qquad (12.23)$$

Now the relation between shear stresses and transverse force, see Figure 12.7, is needed.

$$T = \int_A \tau dA = \int_A G\gamma_{xy} dA \approx \frac{GA\gamma}{\alpha} = GA_s\gamma \qquad (12.24)$$

The parameter α was introduced for enabling the simplification of constant shear over the cross-section. It gives an effective shear area

$$A_s = \frac{A}{\alpha} \qquad (12.25)$$

The value depends on the shape of the cross-section of the beam and is tabulated.

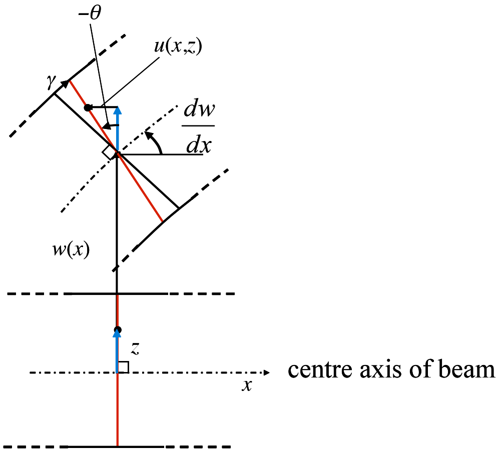


Figure 12.6. Assumed deformation of beam cross-section according to Timoshenko beam theory.

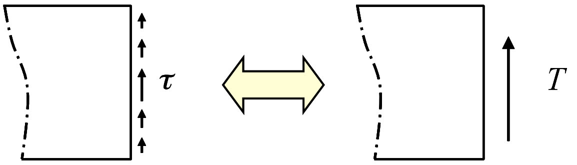


Figure 12.7. Transverse force and shear stresses.

The equilibrium relations for an infinitesimal element, Figure 12.3, are the same as for the Bernoulli beam giving

$$\frac{dM}{dx} = T \qquad (12.26)$$

and

$$\frac{dT}{dx} = -q \qquad (12.27)$$

Combining them gives

$$\frac{d^2M}{dx^2} = -q \qquad (12.28)$$

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 12.8.

We will not formulate the equilibrium equation corresponding to Eq. (12.9) 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.

$$\prod_e = \int_V \left(\frac{1}{2}\sigma\varepsilon + \frac{1}{2}\tau\gamma\right) dAdx \qquad (12.29)$$

Insertion of Eq. (12.22) and Eq. (12.23) in the integral above and integrating over the cross-section gives

$$\prod_e = \int_L \left(\frac{1}{2}EI\left(\frac{d\theta}{dx}\right)^2 + \frac{1}{2}\frac{GA}{\alpha}\gamma^2\right) dx \qquad (12.30)$$

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

$$\Pi = \int_L \frac{1}{2}\begin{bmatrix}\frac{d\theta}{dx} & \gamma\end{bmatrix}\begin{bmatrix}EI & 0\\ 0 & \frac{GA}{\alpha}\end{bmatrix}\begin{bmatrix}\frac{d\theta}{dx}\\ \gamma\end{bmatrix} dx - \int_L qwdx \qquad (12.31)$$

The basic relations are summarised below. The vectors $\mathbf{f}_{ext}$ and $\mathbf{u}$ will be introduced when formulating a finite element in chapter 8.2.

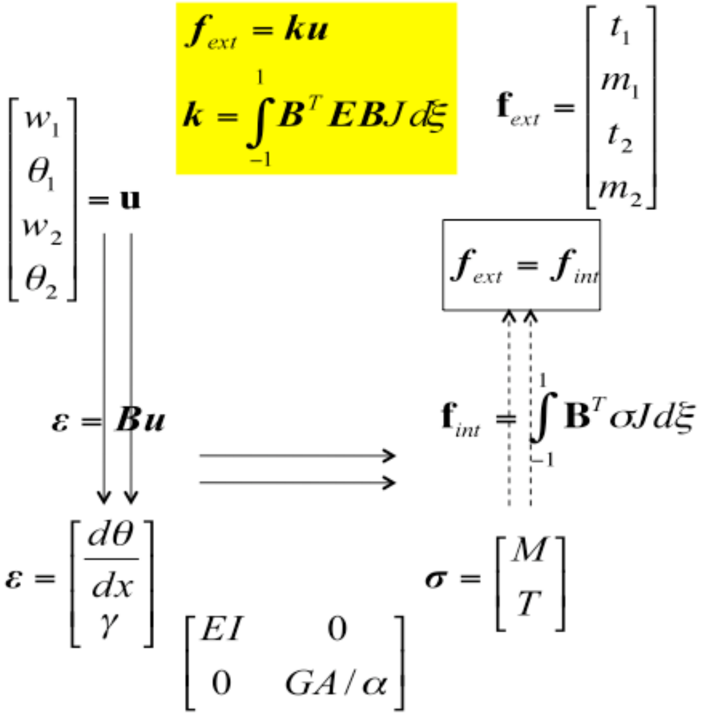

$\boldsymbol{f}_{ext} = \boldsymbol{ku}$
$\boldsymbol{k} = \int_{-1}^{1} \boldsymbol{B}^T \boldsymbol{EBJ} d\xi$
$\mathbf{f}_{ext} = \begin{bmatrix} t_1 \\ m_1 \\ t_2 \\ m_2 \end{bmatrix}$
$\begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{u}$
$\boldsymbol{f}_{ext} = \boldsymbol{f}_{int}$
$\boldsymbol{\varepsilon} = \boldsymbol{Bu}$
$\mathbf{f}_{int} = \int_{-1}^{1} \mathbf{B}^T \sigma J d\xi$
$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{d\theta}{dx} \\ \gamma \end{bmatrix}$
$\begin{bmatrix} EI & 0 \\ 0 & GA/\alpha \end{bmatrix}$
$\boldsymbol{\sigma} = \begin{bmatrix} M \\ T \end{bmatrix}$

Figure 12.8. Tonti diagram for Timoshenko beam theory.

The solution according to Timoshenko beam theory of the deflection of the beam in Figure 12.5 with a square cross-section is

$$w(x)=\frac{PL^3}{EI}\frac{1}{2}\left(\frac{x}{L}\right)^2\left[\frac{1}{3}\left(\frac{x}{L}\right)-1\right]-\frac{P\alpha}{kGA}x$$

where $G=\frac{E}{2(1+\nu)}$ and for $\alpha=\frac{12+11\nu}{10(1+\nu)}$.

This gives the maximum deflection

$$w_{TBT}=-\frac{PL^3}{3EI}-\frac{PL}{GA}\frac{(12+11\nu)}{10(1+\nu)}=-\frac{12PL^3}{3EBH^3}-\frac{PL}{GBH}\frac{(12+11\nu)}{10(1+\nu)}$$

$$w_{LBB}=-\frac{PL^3}{3EI^3}\left[1+\left(\frac{H}{L}\right)^2\frac{(12+11\nu)}{20}\right] \qquad (12.32)$$

12.3 One-dimensional element for heat conduction

The one-dimensional heat conduction equation[23] is written as

$$L(T)=\lambda\frac{d^2T}{dx^2}+\dot{Q}-\rho c\dot{T}=0 \qquad (12.33)$$

where ρ is the density, c is the heat capacity and λ is the heat conductivity. A Tonti diagram for the basic relations is shown in Figure 12.9.

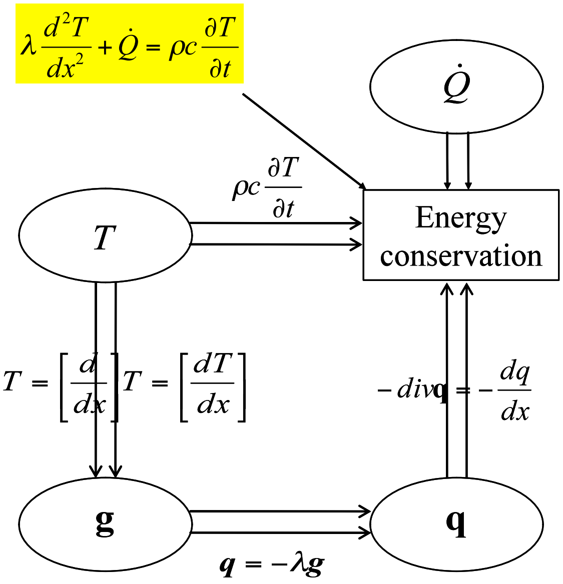

$\lambda\frac{d^2T}{dx^2}+\dot{Q}=\rho c\frac{\partial T}{\partial t}$
$\dot{Q}$
$\rho c\frac{\partial T}{\partial t}$
T
Energy conservation
$\mathbf{g}=\nabla T=\left[\frac{d}{dx}\right]T=\left[\frac{dT}{dx}\right]$
$-div\mathbf{q}=-\frac{dq}{dx}$
$\mathbf{g}$
$\mathbf{q}$
$\mathbf{q}=-\lambda\mathbf{g}$

Figure 12.9. Tonti diagram for heat conduction equation.

The finite element equations corresponding to Figure 12.9 are shown in Figure 12.11. 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

$$T(s)=\mathbf{NT}^e=\begin{bmatrix}N_1 & N_2\end{bmatrix}\begin{bmatrix}T_1\\T_2\end{bmatrix}^e \qquad (12.34)$$

where the finite element shape functions for a two node element, see Figure 12.10, are the same as in section 6.3

$$N_1=\frac{1}{2}(1-s)\quad N_2=\frac{1}{2}(1+s) \qquad (12.35)$$

The element conductivity matrix is

$$\mathbf{k} = \int \mathbf{B}^T \lambda \mathbf{B} J A ds \mathbf{T}^e = \int_{-1}^{1} \lambda \frac{1}{l^e} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \frac{1}{l^e} \begin{bmatrix} -1 & 1 \end{bmatrix} A \frac{l^e}{2} ds \tag{12.36}$$

This gives

$$\mathbf{k} = \frac{\lambda A}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{12.37}$$

The element heat capacity matrix is

$$\mathbf{c} = \int_{-1}^{1} \rho c \mathbf{N}^T \mathbf{N} J A ds = \frac{\rho c A l^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{12.38}$$

The use of a lumped hat capacity matrix can be advantageous sometimes. It is

$$\mathbf{c} = \frac{\rho c A l^e}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{12.39}$$

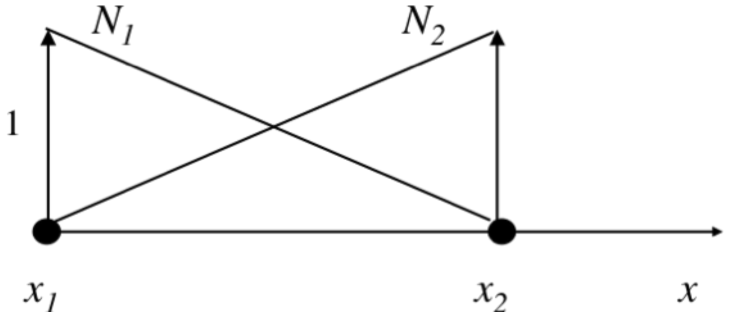
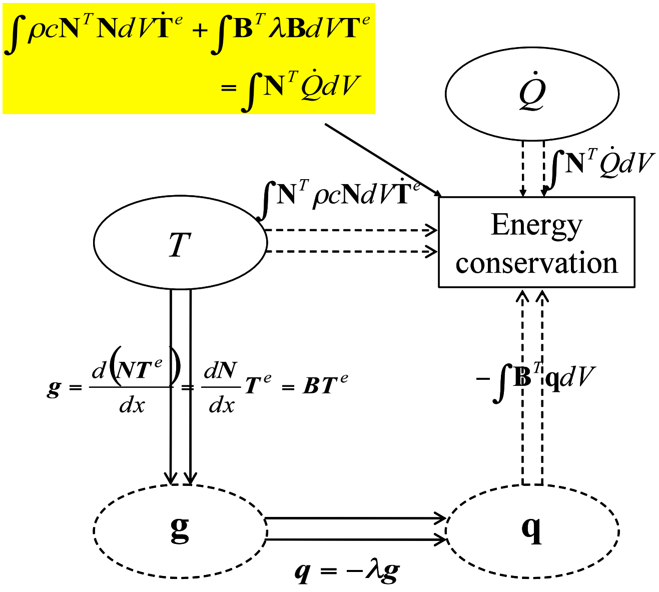


Figure 12.10. Two node element and its linear shape functions N_1 and N_2.



$\int \rho c \mathbf{N}^T \mathbf{N} dV \dot{\mathbf{T}}^e + \int \mathbf{B}^T \lambda \mathbf{B} dV \mathbf{T}^e = \int \mathbf{N}^T \dot{Q} dV$
$\dot{Q}$
$\int \mathbf{N}^T \dot{Q} dV$
$\int \mathbf{N}^T \rho c \mathbf{N} dV \dot{\mathbf{T}}^e$
T
Energy conservation
$\boldsymbol{g} = \frac{d(\boldsymbol{NT}^e)}{dx} = \frac{d\boldsymbol{N}}{dx}\boldsymbol{T}^e = \boldsymbol{BT}^e$
$-\int \mathbf{B}^T \mathbf{q} dV$
$\mathbf{g}$
$\mathbf{q}$
$\boldsymbol{q} = -\lambda \boldsymbol{g}$

Figure 12.11. Tonti diagram for finite element relations for heat conduction.

13 References

1. Hutton, D., *Fundamentals of Finite Element Analysis*. 2004: McGraw-Hill.
2. Belytschko, T., W.K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. 2000: John Wiley & Sons. 650.

3. Fletcher, C., *Computational Galerkin Methods*. Springer Series in Computational Physics, ed. H. Cabannes, et al. 1984, New-York: Springer-Verlag. 320.

4. Onate, E., J. Miquel, and G. Hauke, Stabilized formulation for the advection-diffusion-absorption equation using finite calculus and linear finite elements. Computer Methods in Applied Mechanics and Engineering, 2006. **195**: p. 3926-3946.

5. Levitas, V., et al., *Numerical modelling of martensitic growth in an elastoplastic material.* Philosophical Magazine A, 2002. **82**(3): p. 429-462.

6. Codina, R., Finite Element Formulation for the Numerical Solution of the Convection-Diffusion Equation. Vol. 14. 1993: CIMNE.