Lecture Notes

MATH0086 Computational and Simulation Methods (Part II)

Gert van der Heijden

Fall 2019

Contents

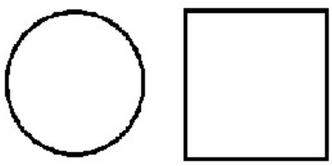
Chapter 1

Preparation

1.1 Finite Elements

Overview. The Finite-Element (FE) method is a way of approximately solving differential equations. Most problems in engineering are governed by (ordinary or partial) differential equations. In many cases these are not solvable by exact methods, so there is a clear need for approximate solution techniques. It is good to realise that the difficulty in solving differential equations is as much in the boundary conditions (BCs) as in the equations themselves (especially in 2D and higher dimensions).

For instance,



Simple to solve using symmetry to reduce problems

Hard to solve a problem with complex geometry

Sketch of the FE Procedure

- **Division**: divide the structure (or domain) into small parts (the **finite elements**) of simple geometry
- **Discretisation**: choose nodes where the solution will be computed (approximately)
- **Interpolation**: interpolate between nodal values to get the solution for the entire structure (or domain). This is to be done such that the solution is continuous across element boundaries (**conformity** or **compatibility**)

Characteristics of the FE Method

- universal, robust (works for any equation and boundary conditions)
- can be successively refined to get solution at any required precision
- can be solved on a computer, not just solving the final set of equations but also the formulation of the approximate equations, the generation of a mesh of finite elements, the application of the BCs, etc.

1.2 1D Approximation Theory Based on Weighted Residuals

Consider the boundary-value problem

$$\begin{aligned} &\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + u + x = 0, \quad x \in [0,1] \\ &u(0) = 0 = u(1), \qquad \text{(Dirichlet BCs)} \end{aligned} \tag{1.1}$$

Assume a trial function (first guess) $\widetilde{u}$ that at least satisfies the BCs; e.g.,

$$\widetilde{u} = ax(1-x). \tag{1.2}$$

Then the error, or residual, is

$$R = \frac{\mathrm{d}^2 \widetilde{u}}{\mathrm{d}x^2} + \widetilde{u} + x. \tag{1.3}$$

We can adjust a to get the "best" approximation of type (1.2), but what is the "best"? The method of **Weighted Residuals** defines a weight function w and demands that the weighted average be 0:

$$I \stackrel{\text{def.}}{=} \int_0^1 wR\mathrm{d}x = 0. \tag{1.4}$$

Note that this gives one equation for the single unknown a, so, given w, we can solve for a.
Next question: what to choose for w?

1.2.1 Three Common Choices

- Collocation Method: $w = \delta(x - x_i)$ (Dirac delta function)
- Least Squares Method: minimise $I = \int R^2 \mathrm{d}x \quad \Rightarrow \quad w = \frac{\mathrm{d}R}{\mathrm{d}a}$
- Galerkin Method: $w =$ trial function $\widetilde{u}$ itself

Collocation Method

Choose $w = \delta(x - x_i)$: all weight concentrated at x_i (gives exact solution at $x = x_i$)

$$I = \int_0^1 \delta(x - x_i) R(x) \mathrm{d}x = R(x_i). \tag{1.5}$$

Take, for instance, $x_i = \frac{1}{2}$ (could be any number in [0,1]),

$$I = R(\frac{1}{2}) = -2a + \frac{1}{4}a + \frac{1}{2} = 0 \quad \Rightarrow \quad a = \frac{2}{7} \approx 0.2857.$$

Least Square Method

$$I = \int_0^1 R^2 \mathrm{d}x \quad \text{then minimise, i.e., set} \quad \frac{\mathrm{d}I}{\mathrm{d}a} = 0. \tag{1.6}$$

Here, $I = \frac{101}{30}a^2 - \frac{11}{6}a + \frac{1}{3} \quad \Rightarrow \quad \frac{\mathrm{d}I}{\mathrm{d}a} = 0 \quad \Rightarrow \quad a = \frac{55}{202} \approx 0.2723.$
Strictly speaking, this is not in weighted residual form, but note that $0 = \frac{\mathrm{d}I}{\mathrm{d}a} = 2\int_0^1 R\frac{\mathrm{d}R}{\mathrm{d}a}\mathrm{d}x$, which is in weighted residual form with weight function

$$w = \frac{\mathrm{d}R}{\mathrm{d}a}. \tag{1.7}$$

Galerkin Method

For w take the trial function itself:

$$w = x(1-x). \tag{1.8}$$

$0 = I = \int_0^1 x(1-x)[-2a + ax(1-x) + x]\mathrm{d}x \quad \Rightarrow \quad a = \frac{5}{18} \approx 0.2778.$

Exact Solution of the Example Problem

Equation (1.1) is an inhomogeneous second-order equation. We can solve it by first solving the corresponding homogeneous equation, $u\prime\prime + u = 0$ (u_{hom}) and adding a particular solution (u_p). The general solution thus takes the form

$$u = u_{hom} + u_p = a\sin x + b\cos x - x. \tag{1.9}$$

By substituting (1.9) into the boundary conditions, we get $b = 0$, $a = \frac{1}{\sin 1}$.
Thus the exact solution is

$$u(x) = \frac{1}{\sin 1}\sin x - x. \tag{1.10}$$

1.2.2 Improving the Approximation

To improve the approximation we can take two terms in the trial solution (such that $\widetilde{u}$ still satisfies the boundary conditions):

$$\widetilde{u} = a_1 x(1-x) + a_2 x^2(1-x). \tag{1.11}$$

Now we need two equations to solve for a_1 and a_2.

Collocation Method

$$w_1 = \delta(x - x_1), \qquad w_2 = \delta(x - x_2), \qquad x_1, x_2 \in (0,1)$$
$$\Rightarrow \quad 2 \text{ eqs } (R(x_1) = 0,\ R(x_2) = 0) \text{ for 2 unknowns } (a_1, a_2).$$

Least Squares Method

$$w_1 = \frac{\partial R}{\partial a_1}, \qquad w_2 = \frac{\partial R}{\partial a_2}$$
$$\Rightarrow \quad 2 \text{ eqs : } \quad \int_0^1 R\frac{\partial R}{\partial a_1}\mathrm{d}x = 0, \qquad \int_0^1 R\frac{\partial R}{\partial a_2}\mathrm{d}x = 0.$$

Galerkin Method

$$w_1 = x(1-x), \qquad w_2 = x^2(1-x)$$
$$\Rightarrow \quad 2 \text{ eqs} : \quad \int_0^1 R w_1 \mathrm{d}x = 0, \qquad \int_0^1 R w_2 \mathrm{d}x = 0.$$

Note that $\widetilde{u}$ could be any function that satisfies the boundary conditions, so, more generally, we could take

$$\widetilde{u} = \sum_{i=1}^{n} a_i x^i (1-x) \qquad (n\text{-term trial function})$$
$$\widetilde{u} = \sum_{i=1}^{n} a_i \sin(n\pi x) \qquad \text{(Fourier series)}$$

1.2.3 General Form

Suppose we have $Lu = f$, where L is a linear differential operator (acting on an arbitrary function u). In our example, $Lu = u'' + u$ and $f = -x$.
The (n-term) trial function is:

$$\widetilde{u} = \sum_{i=1}^{n} a_i \varphi_i, \tag{1.12}$$

where the φ_i are basis trial functions. In our example above (eq. (1.11)), $\varphi_1 = x(1-x)$, $\varphi_2 = x^2(1-x)$.
The residual is:

$$\begin{aligned} R =& L\widetilde{u} - f \\ \xlongequal[\text{linearity}]{(1.12)}& \sum_{i=1}^{n} a_i L\varphi_i - f. \\ \text{Weighted residuals} : \quad & \int_0^1 w_j R \mathrm{d}x = 0 \quad (j = 1, 2, \ldots, n). \end{aligned} \tag{1.13}$$

Least Squares Method

$$\begin{aligned} w_j = \frac{\partial R}{\partial a_j} = L\varphi_j \\ \Rightarrow \quad 0 = \int_0^1 L\varphi_j R \mathrm{d}x = \int_0^1 L\varphi_j \left(\sum_{i=1}^{n} a_i L\varphi_i - f \right) \mathrm{d}x \\ \Leftrightarrow \quad \sum_{i=1}^{n} A_{ji} a_i = f_j \quad \text{(system of } n \text{ linear algebraic equations)} \end{aligned}$$

where $A_{ji} = \int_0^1 L\varphi_j L\varphi_i \mathrm{d}x$, $f_j = \int_0^1 f L\varphi_j \mathrm{d}x$.
Note that the matrix A is symmetric: $A_{ij} = A_{ji}$.

Galerkin Method

$$w_j = \varphi_j$$

$$\xRightarrow{\text{linearity}} \quad 0 = \int_0^1 \varphi_j R \mathrm{d}x = \sum_i^n a_i \int_0^1 \varphi_j L \varphi_i \mathrm{d}x - \int_0^1 f \varphi_j \mathrm{d}x \quad (j = 1, 2, \ldots, n).$$

Write in vector form,

$$A\bar{a} = \bar{f}, \quad \bar{a} = (a_1, \ldots, a_n), \quad \bar{f} = (f_1, \ldots, f_n), \tag{1.14}$$

where $f_i = \int_0^1 f\varphi_i \mathrm{d}x$ and $A_{ij} = \int_0^1 \varphi_i L \varphi_j \mathrm{d}x$.
A_{ij} is not obviously symmetric. It is symmetric if

$$\int_0^1 \varphi_i L \varphi_j \mathrm{d}x = \int_0^1 \varphi_j L \varphi_i \mathrm{d}x \tag{1.15}$$

for all pairs of functions φ_i, φ_j.
We discuss this condition for symmetry of A_{ij} a little further.
Mathematical Fact: there is always an operator L^* such that $\int fLg\mathrm{d}x = \int gL^*f\mathrm{d}x$, for any functions f and g satisfying the (homogeneous) boundary conditions. L^* is called the adjoint operator. If $L^* = L$, then L is called **self-adjoint**.
Thus, A as obtained by the Galerkin method is symmetric if and only if L is self-adjoint, i.e., equation (1.15) holds.

> Analogue for finite-dimensional vectors:
>
> $$x \cdot Ay = y \cdot A^T x \quad (A^T \text{ is the transpose of } A)$$
>
> If $A^T = A$, then A is symmetric.

In our example (1.1),

$$\begin{aligned}
\int_0^1 fLg\mathrm{d}x = &\int_0^1 f(g'' + g)\mathrm{d}x \\
\overset{\text{integration}}{\underset{\text{by parts}}{=\!=\!=}} &\int_0^1 (-f'g' + fg)\mathrm{d}x + [fg']_0^1 \\
= &\int_0^1 (f''g + fg)\mathrm{d}x - [f'g]_0^1 \\
= &\int_0^1 (Lf)g\mathrm{d}x,
\end{aligned}$$

where the boundary conditions have been used. Thus, L is self-adjoint, and the matrix A is symmetric.
Note: $Lu = u' + u$ would not be self-adjoint (it would have the wrong sign) and neither would $Lu = u'' + u'$. However, $Lu = e^x u'' + e^x u' = (e^x u')'$ *is* self-adjoint (e^x is here called an integrating factor).

1.2.4 Example of a Problem with Inhomogeneous BCs

$$u'' + u + x = 0, \qquad u(0) = a, \quad u(1) = b.$$

Such problems are transformed to problems with homogeneous BCs and then solved.
Choose a (simple) function $h(x)$ that satisfies the BCs, $h(0) = a$ and $h(1) = b$. For instance, $h(x) = a + (b - a)x$.
Then the difference $v(x) \overset{\text{def.}}{=\!=} u(x) - h(x)$ satisfies homogeneous BCs: $v(0) = 0 = v(1)$.
Moreover, $v\prime\prime = u\prime\prime - h\prime\prime = u\prime\prime = -u - x = -v - h - x$

$$\Leftrightarrow \quad v\prime\prime + v + h + x = 0, \qquad v(0) = 0 = v(1). \tag{1.16}$$

This is a boundary-value problem for v in standard homogeneous form that can be solved (approximately) using the methods discussed above. Once problem (1.16) is solved for v, we can use the formula $u(x) = v(x) + h(x)$ to find $u(x)$.

1.3 2D Approximation Theory

1.3.1 The Gauss-Green Identity

We will need the analogue of integration by parts (ibp) in 2D.
To derive this we start with the **Gauss Divergence Theorem:**

$$\iint\limits_{\Omega} \mathrm{div}(\vec{v})\, \mathrm{d}x\mathrm{d}y = \int\limits_{\Gamma} \vec{v} \cdot \vec{n}\, \mathrm{d}s. \tag{1.17}$$

Ω is here a 2D domain with boundary Γ, $\vec{v} = v_x\vec{i} + v_y\vec{j}$ is a vector field on Ω, $\vec{n}$ is the outward-pointing unit normal to the boundary, s is a parameter along the boundary and $\mathrm{div}(\vec{v}) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}$. $\mathrm{div}(\vec{v})$ is called the divergence of the vector field $\vec{v}$ and is a measure of sources or sinks of the flow described by $\vec{v}$.
To see this, note that in a 2D infinitesimal element, the total (net) outward flow in the x direction is

$$v_x(x + \mathrm{d}x, y)\mathrm{d}y - v_x(x, y)\mathrm{d}y = \frac{\partial v_x}{\partial x}\mathrm{d}x\mathrm{d}y.$$

Similarly, for the y direction,

$$v_y(x, y + \mathrm{d}y)\mathrm{d}x - v_y(x, y)\mathrm{d}x = \frac{\partial v_y}{\partial y}\mathrm{d}x\mathrm{d}y. \tag{1.18}$$

So, the total outward flow is

$$(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y})\mathrm{d}x\mathrm{d}y.$$

Per unit area this gives:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \overset{\text{def.}}{=\!=} \mathrm{div}(\vec{v}). \tag{1.19}$$

The Gauss divergence theorem (1.17) can therefore be understood as a conservation law:

"net generation in Ω" = "net loss through boundary Γ".

div($\vec{v}$) < 0 (sink) div($\vec{v}$) > 0 (source) div($\vec{v}$) $= 0$

Now let ϕ be a scalar function and $\vec{v}$ be a vector function. Then

$$\begin{aligned}\operatorname{div}(\phi\vec{v}) &= \frac{\partial(\phi v_x)}{\partial x} + \frac{\partial(\phi v_y)}{\partial y} \\ &= \phi\frac{\partial v_x}{\partial x} + v_x\frac{\partial \phi}{\partial x} + \phi\frac{\partial v_y}{\partial y} + v_y\frac{\partial \phi}{\partial y} \\ &= \phi\operatorname{div}(\vec{v}) + \nabla\phi\cdot\vec{v}.\end{aligned}$$

$\boxed{\text{gradient of } \phi\text{: } \nabla\phi = (\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y})}$

Now apply the Gauss divergence theorem to the vector field $\phi\vec{v}$. This gives Green's identity, holding for any ϕ and $\vec{v}$:

$$\iint_{\Omega} \phi\operatorname{div}(\vec{v})\,\mathrm{d}x\mathrm{d}y = \oint_{\Gamma} \phi\vec{v}\cdot\vec{n}\,\mathrm{d}s - \iint_{\Omega} \nabla\phi\cdot\vec{v}\,\mathrm{d}x\mathrm{d}y. \tag{1.20}$$

Finally, take $v_x = \psi$, $v_y = 0$, then

$$\iint_{\Omega} \phi\frac{\partial\psi}{\partial x}\mathrm{d}x\mathrm{d}y = -\iint_{\Omega} \frac{\partial\phi}{\partial x}\psi\mathrm{d}x\mathrm{d}y + \oint_{\Gamma} \phi\psi n_x\mathrm{d}s. \tag{1.21}$$

This is the **Gauss-Green Identity**. It holds for any pair of functions ϕ and ψ and can be interpreted as the integration by parts formula in 2D.

To motivate this, we can formally apply the theorem to a 1D region (i.e., an interval) as follows. Note that the boundary now consists of two points.
The boundary integral becomes

$$\begin{aligned}&\int \phi\psi n_x\mathrm{d}s \\ =&\phi(a)\psi(a)n_x(a) + \phi(b)\psi(b)n_x(b) \\ =&\phi(b)\psi(b) - \phi(a)\psi(a) = [\phi\psi]_a^b.\end{aligned}$$

So the Gauss-Green identity reduces to the familiar integration by parts formula:

$$\int_a^b \phi\psi'\mathrm{d}x = -\int_a^b \phi'\psi\mathrm{d}x + [\phi\psi]_a^b.$$

1.3.2 2D Galerkin Example

Poisson Equation

Poisson equation on rectangle of sides a and b:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -c. \tag{1.22}$$

BCs: $u = 0$ on the boundary Γ (i.e., $u = 0$ at $x = 0, x = a, y = 0, y = b$).
This equation is the simplest second-order problem in 2D. It describes, for instance, the lateral deflection of a membrane under pressure c.

General Form

In operator form: $Lu = f$, with

$$L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad f = -c.$$

Proof that L is self-adjoint:

$$\begin{aligned}
& \iint\limits_{\Omega} \phi \frac{\partial^2 \psi}{\partial x^2} \mathrm{d}x\mathrm{d}y \\
\overset{\text{Gauss-}}{\underset{\text{Green}}{=\!=}}\; & \iint\limits_{\Omega} \psi \frac{\partial^2 \phi}{\partial x^2} \mathrm{d}x\mathrm{d}y + \oint\limits_{\Gamma} \phi \frac{\partial \psi}{\partial x} n_x \mathrm{d}s - \oint\limits_{\Gamma} \psi \frac{\partial \phi}{\partial x} n_x \mathrm{d}s \\
\overset{\oint_\Gamma = 0}{\underset{\text{BCs}}{=\!=}}\; & \iint\limits_{\Omega} \psi \frac{\partial^2 \phi}{\partial x^2} \mathrm{d}x\mathrm{d}y.
\end{aligned}$$

Same for the y direction:

$$\iint\limits_{\Omega} \phi \frac{\partial^2 \psi}{\partial y^2} \mathrm{d}x\mathrm{d}y = \iint\limits_{\Omega} \psi \frac{\partial^2 \phi}{\partial y^2} \mathrm{d}x\mathrm{d}y.$$

We thus have

$$\iint\limits_{\Omega} \phi L \psi \mathrm{d}x\mathrm{d}y = \iint\limits_{\Omega} (L\phi)\psi \mathrm{d}x\mathrm{d}y \tag{1.23}$$

for arbirtray ϕ and ψ, i.e., L is self-adjoint.

Galerkin approximation

We construct a Galerkin approximation for the strong formulation (1.22) of the problem.
Trial solution: $\widetilde{u}(x, y) = \sum_{i=1}^{n} a_i \varphi_i(x, y)$, where the φ_i are basis trial functions that satisfy the (homogeneous) BCs.
Residual: $R = L\widetilde{u} + c$.
Galerkin approximation: $\int \varphi_j R \mathrm{d}x\mathrm{d}y = 0$, for $j = 1, \ldots, n$ leads to a set of equations $A\bar{a} = \bar{f}$, where $A_{ij} = \iint \varphi_j L \varphi_i \mathrm{d}x\mathrm{d}y$, $\bar{a} = (a_1, \ldots, a_n)$, $\bar{f} = (f_1, \ldots, f_n)$ and $f_i = \iint -c\varphi_i \mathrm{d}x\mathrm{d}y$ (in our case).
Taking $n = 1$ and $\varphi_1 = xy(x - a)(y - b)$, we get $Aa_1 = f_1$ with

$$\begin{aligned}
A &= \iint\limits_{\Omega} \varphi_1 L \varphi_1 \mathrm{d}x\mathrm{d}y = \int_0^b \int_0^a xy(x-a)(y-b)[2y(y-b) + 2x(x-a)]\mathrm{d}x\mathrm{d}y \\
&= \cdots = -\frac{a^3 b^3}{90}(a^2 + b^2),
\end{aligned}$$

$$f_1 = -c \iint\limits_{\Omega} \varphi_1 \mathrm{d}x\mathrm{d}y = -c\int_0^a \int_0^b xy(x-a)(y-b)\mathrm{d}x\mathrm{d}y = \cdots = -\frac{c}{36}a^3b^3$$

$$\Rightarrow \quad a_1 = \frac{f_1}{A} = \frac{5c}{2(a^2+b^2)}.$$

We thus find the approximation $\widetilde{u} = \frac{5c}{2(a^2+b^2)}xy(x-a)(y-b)$.

1.4 Weak Formulation

1.4.1 1D

Consider again the example problem

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + u + x = 0, \qquad u(0) = 0 = u(1). \tag{1.24}$$

This so-called **strong formulation** requires $\frac{\mathrm{d}^2u}{\mathrm{d}x^2}$ to exist. This is fine for the exact solution which will be twice differentiable, but we are interested in approximations and for those the existence of a second derivative is overly restrictive. For instance, for piecewise linear approximations (trial solutions) $\widetilde{u}$ (which will be important in FE), $\frac{\mathrm{d}^2\widetilde{u}}{\mathrm{d}x^2}$ is not defined, so we cannot use the weighted residual method in the form used in Section 1.2. Fortunately, this problem can be easily solved by using the so-called weak formulation obtained as follows.

Multiply the equation by an arbitrary function v and integrate:

$$\begin{aligned} &\int_0^1 (\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + u + x)v\mathrm{d}x = 0 \\ \overset{\text{ibp}}{\Longleftrightarrow} &\int_0^1 (-\frac{\mathrm{d}u}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x} + uv + xv)\mathrm{d}x + [\frac{\mathrm{d}u}{\mathrm{d}x}v]_0^1 = 0. \end{aligned} \tag{1.25}$$

The function v in this context is called a **test function**. If u solves (1.25) for *all* possible test functions v, then we expect u also to solve (1.24). In fact, it is sufficient to consider all v that satisfy the *homogeneous* form of any Dirichlet boundary conditions. One then has the equivalence between the above strong formulation (the BVP, (1.24)) and the **weak formulation** defined as

$$\begin{cases} \int_0^1 (-\frac{\mathrm{d}u}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x} + uv + xv)\mathrm{d}x = 0 \\ u(0) = u(1) = 0, \quad \text{for all } v \text{ with } v(0) = 0 = v(1). \end{cases} \tag{1.26}$$

This weak form of the problem only involves first derivatives. In fact, (1.26) makes sense even for functions u (and v) that are merely piecewise differentiable since we can break up the integral into a sum of integrals over any finite number of subintervals. Such u (e.g., piecewise linear functions) are called weakly differentiable. Thus the weak and strong formulations are equivalent for functions that are sufficiently smooth (at least twice differentiable), but the weak formulation requires only the first derivative and therefore allows a much wider class of trial functions to be used in constructing approximate solutions.

Exercise: Show that the weak formulation (1.26) of our example problem

(1.24) for the trial solution $\tilde{u} = a_1 x(1-x) + a_2 x^2(1-x)$ and using the Galerkin method (with $v_1 = x(1-x)$ and $v_2 = x^2(1-x)$) gives the same solution as the 'strong' Galerkin method in Section 1.2.2.

1.4.2 2D

Poisson equation (strong formulation):

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \tag{1.27}$$

Boundary conditions:

$$\begin{cases} u = 0 \quad \text{on} \quad \Gamma_1 \\ \nabla u \cdot \vec{n} = g \quad \text{on} \quad \Gamma_2 \quad \text{(given flux)} \end{cases} \tag{1.28}$$

After Gauss-Green (with test function v):

$$\begin{cases} \displaystyle\int_\Omega \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) \mathrm{d}x\mathrm{d}y + \int_\Omega f v \,\mathrm{d}x\mathrm{d}y - \int_{\Gamma_2} vg \,\mathrm{d}s = 0 \\ u = 0 \quad \text{on} \quad \Gamma_1 \quad \text{(for all } v \text{ with } v = 0 \text{ on } \Gamma_1) \end{cases} \tag{1.29}$$

This is the weak formulation.

Chapter 2

Finite Elements in 1D

So far we have constructed trial solutions on the full domain. Since trial functions are required to satisfy any Dirichlet boundary conditions, this will in general be a hard task in 2D and higher dimensions. The key idea of FE is to subdivide the domain into finite elements of regular shape on which boundary conditions can easily be posed. We consider the 1D case first.

2.1 1D Elements

2.1.1 2-Node Element

A 1D element is simply an interval. We label the nodes i and j and denote the unknown field variable that we are interested in by T (think of temperature in a heat flow problem). Continuity of T across element boundaries, i.e., interval end points x_i and x_j, will be guaranteed if we place nodes at these end points and construct a trial function $\tilde{T} = \sum_k a_k \phi_k$ that satisfies the consistency conditions $\tilde{T}(x_i) = T_i$ and $\tilde{T}(x_j) = T_j$. Here T_i and T_j are the nodal values of T, whatever they are (they will be fixed later by using the equations). For a 2-node element this means that we can allow for 2 parameters in the trial function. In finite elements one simply takes the polynomial (dropping the tilde) $T(x) = \alpha_1 + \alpha_2 x$, corresponding to the choice $\phi_1 = 1$ and $\phi_2 = x$.

2-node element

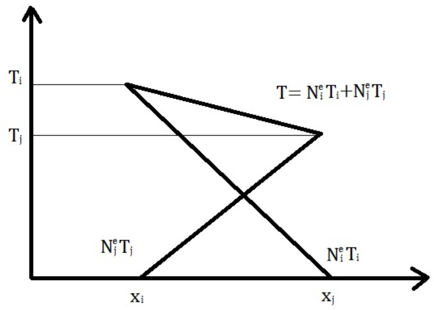
We thus have the following two equations for the two unknowns α_i:

$$\left.\begin{aligned} \alpha_1 + \alpha_2 x_i &= T_i \\ \alpha_1 + \alpha_2 x_j &= T_j \end{aligned}\right\} \qquad \Rightarrow \qquad \begin{aligned} \alpha_2 &= \frac{T_j - T_i}{L} \quad (L = x_j - x_i) \\ \alpha_1 &= T_i - \frac{T_j - T_i}{L} x_i. \end{aligned}$$

Then we can write

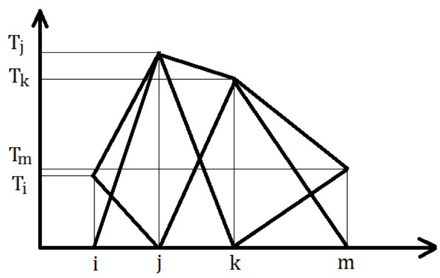
$$T(x) = N_i^e(x)T_i + N_j^e(x)T_j, \tag{2.1}$$

where $N_i^e(x) = -\frac{x-x_j}{L}$, $N_j^e(x) = \frac{x-x_i}{L}$ (**element shape functions**).
Note that in this form T is expressed as interpolation between nodal values, with the x-variation given by the shape functions.



linear interpolation between nodal values

2-node element interpolation



on mesh of three 2-node elements

It is useful to define **global shape functions** N_i, N_j, etc., defined over the full domain (i.e., for all x) and attached to the nodes (rather than to the elements).
We can then also write, on the first element,

$T(x) = N_i(x)T_i + N_j(x)T_j$,

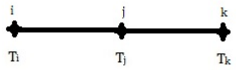
and on a mesh of three 2-node elements,

$T(x) = N_i(x)T_i + N_j(x)T_j + N_k(x)T_k + N_m(x)T_m$.

These 2-node global shape functions N_i are also called tent functions, for obvious reasons.
Important property: $N_p(x_q) = \delta_{pq}$ (Kronecker symbol: $\delta_{pq} = 1$ if $p = q$ and 0 if $p \neq q$), i.e., $N_p = 1$ at its own node and 0 at all other nodes.

2.1.2 Higher-order Elements

3-node (quadratic) element



3-node element

approximation: $T = \alpha_1 + \alpha_2 x + \alpha_3 x^2$
consistency: $T(x_i) = T_i, \dots$

$$\begin{aligned} T_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 x_i^2 \\ T_j &= \alpha_1 + \alpha_2 x_j + \alpha_3 x_j^2 \\ T_k &= \alpha_1 + \alpha_2 x_k + \alpha_3 x_k^2 \end{aligned}$$

or in matrix form

$$\begin{pmatrix} 1 & x_i & x_i^2 \\ 1 & x_j & x_j^2 \\ 1 & x_k & x_k^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} T_i \\ T_j \\ T_k \end{pmatrix}.$$

Solve by using Cramer's Rule:

$$\alpha_1 = \frac{\begin{vmatrix} T_i & x_i & x_i^2 \\ T_j & x_j & x_j^2 \\ T_k & x_k & x_k^2 \end{vmatrix}}{\begin{vmatrix} 1 & x_i & x_i^2 \\ 1 & x_j & x_j^2 \\ 1 & x_k & x_k^2 \end{vmatrix}}, \quad \alpha_2 = \frac{\begin{vmatrix} 1 & T_i & x_i^2 \\ 1 & T_j & x_j^2 \\ 1 & T_k & x_k^2 \end{vmatrix}}{\begin{vmatrix} 1 & x_i & x_i^2 \\ 1 & x_j & x_j^2 \\ 1 & x_k & x_k^2 \end{vmatrix}}, \quad \alpha_3 = \frac{\begin{vmatrix} 1 & x_i & T_i \\ 1 & x_j & T_j \\ 1 & x_k & T_k \end{vmatrix}}{\begin{vmatrix} 1 & x_i & x_i^2 \\ 1 & x_j & x_j^2 \\ 1 & x_k & x_k^2 \end{vmatrix}}$$

$$\Rightarrow T(x) = N_i^e(x) T_i + N_j^e(x) T_j + N_k^e T_k,$$

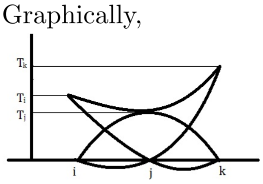
where the element shape functions are

$$\begin{aligned} N_i^e(x) &= \frac{2}{L^2}(x - x_j)(x - x_k) \qquad (L = x_k - x_i) \\ N_j^e(x) &= -\frac{4}{L^2}(x - x_i)(x - x_k) \\ N_k^e(x) &= \frac{2}{L^2}(x - x_i)(x - x_j) \end{aligned}$$

(we have assumed here that node j is halfway between nodes i and k).
Note

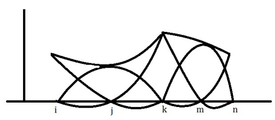
$$N_i^e = \begin{cases} 1 & \text{at node i} \\ 0 & \text{at all other nodes} \end{cases}$$

Graphically,

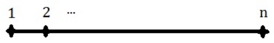


3-node element interpolation

On a mesh of two elements:



Two 3-node element interpolation

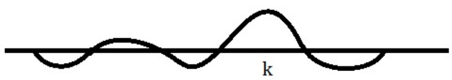


$$T = \alpha_1 + \alpha_2 x + \cdots + \alpha_n x^{n-1}$$

n-node element

Generally, for an n-node element we have $T = N_i^e T_i + \cdots + N_n^e T_n$ and we can construct the shape functions N_i^e using the same procedure. However, there is a simpler way since we know that we require $N_i^e(x_j) = \delta_{ij}$. We can therefore use the **Lagrange interpolation formula**

$$P_k^{n-1}(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_1)(x_k - x_2)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)} \quad (2.2)$$



Lagrange interpolation

(polynomial through all points x_i of a grid except x_k where the value is 1). Indeed, one can show that $N_k^e(x) = P_k^{n-1}(x)$ for $k = 1, ..., n$ (i.e., all 1D shape functions are given by the Lagrange formula).
Check, for $n = 2$:

$$P_1^1(x) = \frac{x - x_2}{x_1 - x_2} = -\frac{x - x_2}{L} = N_1^e(x) \qquad (L = x_2 - x_1)$$
$$P_2^1(x) = \frac{x - x_1}{x_2 - x_1} = \frac{x - x_1}{L} = N_2^e(x).$$

Exercise: Show that for $n = 3$ we get the quadratic shape functions for a 3-node element computed above.

2.1.3 General properties of global shape functions

- $N_i(x_j) = \delta_{ij}$ (i.e., shape function $= 1$ at its own node, $= 0$ at all other nodes)
- $N_i(x) \neq 0$ only on those elements that contain node i (they are therefore also sometimes called localised basis functions)
- $T = \alpha_1 + \alpha_2 x + \cdots + \alpha_n x^{n-1}$ (approximation)$= \sum_{i=1}^n N_i(x) T_i$ (interpolation in terms of shape functions).
  Now take $T = T_0 =$ const; then $\alpha_1 = T_0$, $\alpha_i = 0$ for $i > 1$, and $T_0 = \sum_{i=1}^n N_i(x) T_0$ and hence

  $$\sum_{i=1}^n N_i(x) = 1, \quad \text{for all } x \qquad \text{(partition of unity property).}$$

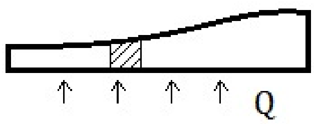
2.2 1D Examples

2.2.1 Steady-state Heat Flow

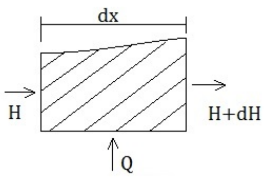
We have constructed trial solutions on a 1D mesh in the special form $T = \sum_{i=1}^{n} N_i(x)T_i$, where the basis functions are the global shape functions N_i and the coefficients T_i are the nodal values of the unknown field variable T. To find these coefficients we need to use the equations for the problem at hand. Here we consider 1D heat flow.

FE Equation for 1D Heat Flow

Consider a long thin fin (as used to cool machinery or storage tanks). We first derive the equation for heat flow in this fin.



long and slender $\Rightarrow$ 1D body
Q = heat input per unit length and unit time (from surrounding fluid)
H = heat flow per unit time



Energy balance:

$$\begin{aligned} & H + Q\mathrm{d}x = H + \mathrm{d}H \\ \Rightarrow \quad & \tfrac{\mathrm{d}H}{\mathrm{d}x} = Q \end{aligned}$$

The heat flux q is defined by

$$\begin{aligned} H(x) &= A(x)q(x) \\ A &= \text{cross-sectional area} \\ q &= \text{flux} = \text{energy passing through unit area per unit time} \end{aligned}$$

According to Fourier's Law (constitutive law), $q = -k\frac{\mathrm{d}T}{\mathrm{d}x}$, where k is the thermal conductivity (i.e., the heat flux is proportional to the temperature gradient). Combining,

$$\frac{\mathrm{d}}{\mathrm{d}x}(Ak\frac{\mathrm{d}T}{\mathrm{d}x}) + Q = 0 \quad \text{(\textbf{1D heat equation}, stationary version)} \tag{2.3}$$

Typical BCs: $T(0) = T_0$, $q(L) = q_L$ (for insulated end, $q = 0$).

Weak formulation

$$
\begin{aligned}
0 &= \int_0^L \Big(\frac{\mathrm{d}}{\mathrm{d}x}\big(Ak\frac{\mathrm{d}T}{\mathrm{d}x}\big) + Q\Big)v\mathrm{d}x \\
&= \int_0^L \Big(-Ak\frac{\mathrm{d}T}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x} + Qv\Big)\mathrm{d}x + \Big[Ak\frac{\mathrm{d}T}{\mathrm{d}x}v\Big]_0^L \\
&= \int_0^L \Big(-Ak\frac{\mathrm{d}T}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x} + Qv\Big)\mathrm{d}x - Aq_L v(L),
\end{aligned}
$$

because test functions v satisfy the homogeneous form of any Dirichlet BCs (here, $T(0) = T_0$ and thus $v(0) = 0$).

It is however convenient and customary in FE to suspend the application of the Dirichlet BCs and to work with

$$\int_0^L \Big(-Ak\frac{\mathrm{d}T}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x} + Qv\Big)\mathrm{d}x - [Aqv]_0^L = 0$$

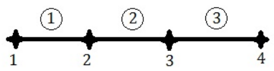
and with the trial function $T(x) = \sum_{j=1}^n N_j(x)T_j$ (with N_j the global shape functions), in which all T_i are considered unknown. The boundary conditions will then be applied afterwards.
We need n equations for the unknown T_i. Following the Galerkin weighted residual method we now choose the basis functions N_i as test/weight functions. Thus: $v_i = N_i \quad (i = 1, \ldots, n)$. Then

$$
\begin{aligned}
&\sum_{j=1}^n \int_0^L \frac{\mathrm{d}N_j}{\mathrm{d}x}Ak\frac{\mathrm{d}N_i}{\mathrm{d}x}\mathrm{d}x T_j = \int_0^L QN_i\mathrm{d}x - [N_iAq]_0^L \\
&\text{or} \quad \sum_{j=1}^n K_{ij}T_j = f_{li} + f_{bi}, \\
\text{where} \quad & K_{ij} = \int_0^L \frac{\mathrm{d}N_i}{\mathrm{d}x}Ak\frac{\mathrm{d}N_j}{\mathrm{d}x}\mathrm{d}x \qquad (\textbf{stiffness matrix}) \\
& f_{li} = \int_0^L QN_i\mathrm{d}x \qquad (\textbf{load vector}) \\
& f_{bi} = -[N_iAq]_0^L \qquad (\textbf{boundary vector}).
\end{aligned}
$$

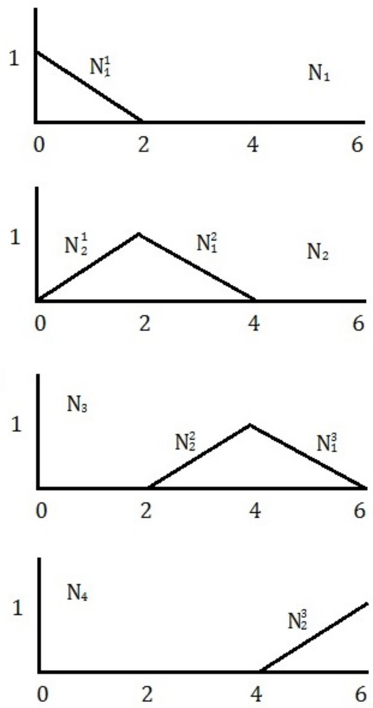
The FE equation thus takes the form

$$Ku = f_l + f_b. \tag{2.4}$$

3-Element Model for Heat Flow in a Uniform Fin



A, k, Q constant.
Take a mesh of three 2-node elements.



Element shape functions ($l = 2$):

$$N_1^1 = \frac{-(x - x_2)}{l} = \frac{2 - x}{2}$$
$$N_2^1 = \frac{x - x_1}{l} = \frac{x}{2}$$
$$N_1^2 = \frac{-(x - x_3)}{l} = \frac{4 - x}{2}$$
$$N_2^2 = \frac{x - x_2}{l} = \frac{x - 2}{2}$$
$$N_1^3 = \frac{-(x - x_4)}{l} = \frac{6 - x}{2}$$
$$N_2^3 = \frac{x - x_3}{l} = \frac{x - 4}{2}$$

Stiffness matrix:

$$K_{11} = \int_0^6 \frac{\mathrm{d}N_1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_1}{\mathrm{d}x} \mathrm{d}x = \int_0^2 \frac{\mathrm{d}N_1^1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_1^1}{\mathrm{d}x} \mathrm{d}x = \frac{1}{2} Ak$$
$$K_{12} = \int_0^6 \frac{\mathrm{d}N_1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_2}{\mathrm{d}x} \mathrm{d}x = \int_0^2 \frac{\mathrm{d}N_1^1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_2^1}{\mathrm{d}x} \mathrm{d}x = -\frac{1}{2} Ak$$
$$K_{13} = \int_0^6 \frac{\mathrm{d}N_1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_3}{\mathrm{d}x} \mathrm{d}x = 0 \quad (N_1 = 0 \text{ on } \textcircled{2}\textcircled{3}, N_3 = 0 \text{ on } \textcircled{1})$$
$$K_{14} = 0$$
$$K_{22} = \int_0^6 \frac{\mathrm{d}N_2}{\mathrm{d}x} Ak \frac{\mathrm{d}N_2}{\mathrm{d}x} \mathrm{d}x = \int_0^2 \frac{\mathrm{d}N_2^1}{\mathrm{d}x} Ak \frac{\mathrm{d}N_2^1}{\mathrm{d}x} \mathrm{d}x + \int_2^4 \frac{\mathrm{d}N_1^2}{\mathrm{d}x} Ak \frac{\mathrm{d}N_1^2}{\mathrm{d}x} \mathrm{d}x = Ak$$
$$K_{23} = \int_0^6 \frac{\mathrm{d}N_2}{\mathrm{d}x} Ak \frac{\mathrm{d}N_3}{\mathrm{d}x} \mathrm{d}x = \int_2^4 \frac{\mathrm{d}N_1^2}{\mathrm{d}x} Ak \frac{\mathrm{d}N_2^2}{\mathrm{d}x} \mathrm{d}x = -\frac{1}{2} Ak$$

We find

$$K = \frac{1}{2} Ak \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}.$$

Load vector:

$$
\begin{aligned}
f_{l1} &= \int_0^6 N_1 Q \mathrm{d}x = Q \int_0^2 N_1^1 \mathrm{d}x = Q \\
f_{l2} &= \int_0^6 N_2 Q \mathrm{d}x = Q \int_0^2 N_2^1 \mathrm{d}x + Q \int_2^4 N_1^2 \mathrm{d}x = 2Q \\
f_{l3} &= \int_0^6 N_3 Q \mathrm{d}x = Q \int_2^4 N_2^2 \mathrm{d}x + Q \int_4^6 N_1^3 \mathrm{d}x = 2Q \\
f_{l4} &= \int_0^6 N_4 Q \mathrm{d}x = Q \int_4^6 N_2^3 \mathrm{d}x = Q
\end{aligned}
$$

$$
f_l = Q \begin{pmatrix} 1 & 2 & 2 & 1 \end{pmatrix}^{\mathrm{T}}.
$$

Boundary vector:

$$
f_{bi} = -[N_i A q]_0^6 = -(N_i A q)|_{x=6} + (N_i A q)|_{x=0}.
$$

Note that $x = 0$ corresponds to node 1 where $N_1 = 1$ and $N_2, N_3, N_4 = 0$. Similarly, $x = 6$ corresponds to node 4, where $N_4 = 1$ and $N_1, N_2, N_3 = 0$. Thus

$$
f_b = \begin{pmatrix} Aq(0) & 0 & 0 & -Aq(6) \end{pmatrix}^{\mathrm{T}}.
$$

FE equation:

$$
\frac{1}{2} Ak \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} = Q \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} Aq(0) \\ 0 \\ 0 \\ -Aq(6) \end{pmatrix}
$$

BCs:

$$
\begin{aligned}
T(0) = T_0 \Rightarrow T_1 = T_0 \\
q(6) = q_L.
\end{aligned}
$$

By suspending the T_0 boundary condition initially, we now have the extra first equation, which can be used to compute $q(0)$ in addition to the T_i.
The solution method is then:

- solve last 3 equations for T_2, T_3, T_4
- having found T_i, use the first equation to find the flux $q(0)$

Reduced 3×3 system (note extra term on right-hand side!):

$$
\frac{1}{2} Ak \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} T_2 \\ T_3 \\ T_4 \end{pmatrix} = \begin{pmatrix} 2Q + \frac{1}{2} AkT_0 \\ 2Q \\ -QAq_L \end{pmatrix}.
$$

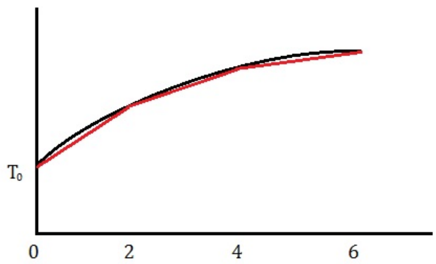
Solution:

$$
\begin{aligned}
T_2 &= (AkT_0 - 2Aq_L + 10Q)/Ak \\
T_3 &= (AkT_0 - 4Aq_L + 16Q)/Ak \\
T_4 &= (AkT_0 - 6Aq_L + 18Q)/Ak
\end{aligned}
$$

The 1st equation of the FE system then gives

$$q(0) = \frac{1}{2}k(T_0 - T_2) - \frac{Q}{A} = \cdots = q_L - \frac{6Q}{A} \qquad (\frac{6Q}{A} \text{ is the heat input}).$$

Graphically



$$T = \sum_{i=1}^{n} N_i(x) T_i$$

Linear interpolation between nodal values

In this case, the nodal values T_i are in fact exact. This is not a general rule (although it is true in 1D for arbitrary Q as long as Ak is constant).

Exact solution

$$\begin{aligned}
& & \frac{\mathrm{d}^2T}{\mathrm{d}x^2} = -\frac{Q}{Ak} \Rightarrow \frac{\mathrm{d}T}{\mathrm{d}x} = -\frac{Qx}{Ak} + c \\
\text{BC:} & & q(6) = -k\frac{\mathrm{d}T}{\mathrm{d}x}(6) = q_L \\
\Rightarrow & & c = -\frac{q_L}{k} + \frac{6Q}{Ak} \\
& & \frac{\mathrm{d}T}{\mathrm{d}x} = -\frac{q_L}{k} - \frac{Q}{Ak}(x-6) \\
& & T(x) = -\frac{q_L}{k}x - \frac{Q}{Ak}(\frac{1}{2}x^2 - 6x) + d \\
\text{BC:} & & T(0) = T_0 \Rightarrow d = T_0 \\
\Rightarrow & & T(x) = T_0 - \frac{q_L x}{k} - \frac{Q}{2Ak}x(x-12).
\end{aligned}$$

At $x = 2, 4, 6$ this gives T_2, T_3, T_4 at the nodes.

2.2.2 General properties of the stiffness matrix

$$K_{ij} = \int_0^L \frac{\mathrm{d}N_i}{\mathrm{d}x} Ak \frac{\mathrm{d}N_j}{\mathrm{d}x} \mathrm{d}x$$

- Symmetric: $K_{ij} = K_{ji}$
- Banded. For instance, tridiagonal for linear 2-node element, because $N_i N_j = 0$ if $|i-j| > 1$ (tent functions)
- Singular: $\det K = 0$
  K does not depend on Q, so consider the homogeneous equation with $Q = 0$: $\frac{\mathrm{d}}{\mathrm{d}x}(Ak\frac{\mathrm{d}T}{\mathrm{d}x}) = 0$. This equation has the constant solution $T(x) = T_0 = \text{const.}$ for any T_0. The FE equation will be $Ku = 0$, with solution

$u = (T_0, ..., T_0)^T$. But for $Ku = 0$ to have a non-trivial ($\neq 0$) solution u requires that $\det K = 0$.
Reason: we have not applied BCs yet, which will pick out a unique solution.

- Rows and columns sum up to zero:
from the partition of unity property $\sum_{i=1}^n N_i(x) = 1$,

$$\sum_{i=1}^n K_{ij} = \int_0^L \frac{\mathrm{d}}{\mathrm{d}x}(\sum_{i=1}^n N_i(x)) Ak \frac{\mathrm{d}N_j}{\mathrm{d}x} \mathrm{d}x = 0.$$

Similarly, $\sum_{j=1}^n K_{ij} = 0$.

Chapter 3

Finite Elements in 2D

3.1 2D Elements

3.1.1 3-Node Triangular Element

Approximation (3 nodes so we can allow 3 coefficients):

$$T(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y = B\alpha$$
$$B = \begin{pmatrix} 1 & x & y \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}.$$

For consistency:

$$\begin{aligned} T(x_1,y_1) &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 = T_1, \\ T(x_2,y_2) &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 = T_2, \\ T(x_3,y_3) &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 = T_3. \end{aligned}$$

In matrix form:

$$\begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \quad \text{or} \quad C\alpha = u, \quad u = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}.$$

We want: $T = N_1^e T_1 + N_2^e T_2 + N_3^e T_3 = Nu, \quad N = (N_1^e, N_2^e, N_3^e).$
Hence

$$T = B\alpha = BC^{-1}u = Nu \quad \Rightarrow \quad N = BC^{-1}.$$

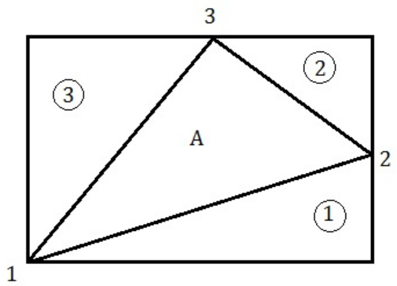
Shape functions are thus:

$$\begin{aligned} N_1^e &= \frac{1}{2A}[x_2 y_3 - x_3 y_2 + (y_2 - y_3)x + (x_3 - x_2)y], \\ N_2^e &= \frac{1}{2A}[x_3 y_1 - x_1 y_3 + (y_3 - y_1)x + (x_1 - x_3)y], \\ N_3^e &= \frac{1}{2A}[x_1 y_2 - x_2 y_1 + (y_1 - y_2)x + (x_2 - x_1)y], \end{aligned}$$

where $A = \frac{1}{2}\det C$.
Note that $N_i^e(x_j, y_j) = \delta_{ij}$, $\sum_{i=1}^{3} N_i^e = 1$.

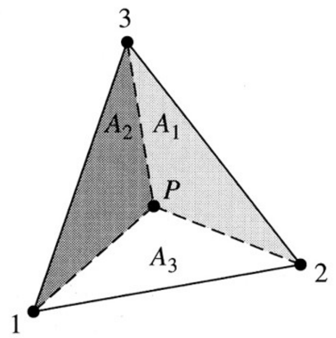
Claim: A is the area of the triangle.

Proof:



$$
\begin{aligned}
A &= (x_2 - x_1)(y_3 - y_1) \\
&\quad - \text{area}① - \text{area}② - \text{area}③ \\
&= \cdots = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2}\det C
\end{aligned}
$$

The N_i^e have a geometrical interpretation. To see this, introduce the **area coordinates** η_1, η_2, η_3 as follows:



$$
\eta_1 = \frac{\text{area(P23)}}{\text{area(123)}}
$$
$$
\eta_2 = \frac{\text{area(P31)}}{\text{area(123)}}
$$
$$
\eta_3 = \frac{\text{area(P12)}}{\text{area(123)}}
$$

Note that $\eta_1 + \eta_2 + \eta_3 = 1$. Moreover, at node 1, $\eta_1(x_1, y_1) = 1$, while $\eta_2 = \eta_3 = 0$. Same properties as shape function N_1^e! Indeed, we can show that $N_i^e = \eta_i$.

Exercise: Show this.

$$
\eta_1 = \frac{\text{area(P23)}}{\text{area(123)}} = \frac{\frac{1}{2}\det \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}}{A} = \cdots = N_1^e.
$$

By direct calculation one easily establishes the inverse relationship between Cartesian and (redundant) area coordinates:

$$
\begin{cases} x = x_1\eta_1 + x_2\eta_2 + x_3\eta_3, \\ y = y_1\eta_1 + y_2\eta_2 + y_3\eta_3. \end{cases}
$$

Area coordinates are useful to evaluate integrals of type $\int x^n y^m \mathrm{d}x\mathrm{d}y$ (as appear in K and f_l).
Idea: Transform $(x, y) \to (\eta_1, \eta_2, \eta_3)$ and use the **integration formula**

$$\iint_A \eta_1^m \eta_2^n \eta_3^p \,\mathrm{d}x\mathrm{d}y = \frac{m!\,n!\,p!}{(m+n+p+2)!}\,(2A). \tag{3.1}$$

Exercise: Use this formula to compute the following integrals over an arbitrary triangle: $\int \mathrm{d}x\mathrm{d}y$, $\int x\,\mathrm{d}x\mathrm{d}y$, $\int xy\,\mathrm{d}x\mathrm{d}y$.

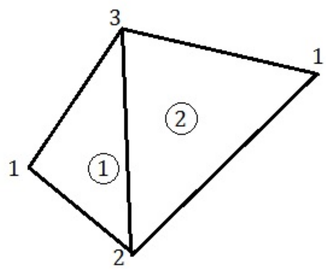
General property of 3-node triangular elements: $N_i^e = 0$ along sides not containing node i.

Proof:

Take node 1 and let side 2-3 be represented as $y = c_1 + c_2 x$ (straight line). Then $N_1^e(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y = a + bx$ for some coefficients a and b. By the $N_i^e(x_j, y_j) = \delta_{ij}$ property of shape functions we then have $a + bx_2 = 0$ and also $a + bx_3 = 0$. This implies $a = 0$ and $b = 0$ and hence $N_1^e = 0$ along the side 2-3.

Conformity

A mesh is called **conforming** (or compatible) if it guarantees continuity of the solution $T(x, y)$ across element boundaries.
Consider two neighbouring elements in a mesh:



We already know from the above property that $N_1^1 = 0 = N_1^2$ along the common side 2-3.
Now consider N_2^1 and N_2^2. We have along side 2-3

$$N_2^1(x, y) = \beta_1 + \beta_2 x + \beta_3 y \xlongequal[\text{side 2-3}]{y=c_1+c_2x} a_1 + b_1 x \quad \text{(for some } a_1, b_1\text{)},$$
$$N_2^2(x, y) = \gamma_1 + \gamma_2 x + \gamma_3 y \xlongequal[\text{side 2-3}]{y=c_1+c_2x} a_2 + b_2 x \quad \text{(for some } a_2, b_2\text{)}.$$

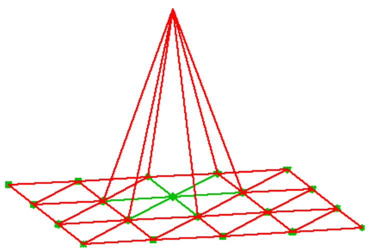
But $N_2^1(x_2, y_2) = 1 = N_2^2(x_2, y_2)$ and $N_2^1(x_3, y_3) = 0 = N_2^2(x_3, y_3)$

$$\Rightarrow \left.\begin{aligned} a_1 + b_1 x_2 = 1 \\ a_2 + b_2 x_2 = 1 \end{aligned}\right\} \Rightarrow (a_1 - a_2) + (b_1 - b_2)x_2 = 0$$

$$\left.\begin{aligned} a_1 + b_1 x_3 = 0 \\ a_2 + b_2 x_3 = 0 \end{aligned}\right\} \Rightarrow (a_1 - a_2) + (b_1 - b_2)x_3 = 0$$

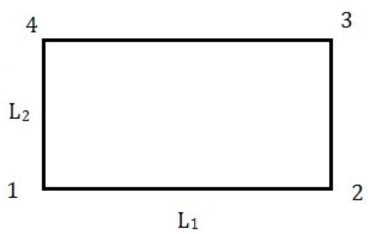
$$x_2 \neq x_3 \quad \Rightarrow \quad b_1 = b_2 \quad \Rightarrow \quad a_1 = a_2 \quad \Rightarrow \quad N_2^1 = N_2^2 \;\; \text{(on side 2-3).}$$

Similarly, $N_3^1 = N_3^2$ on side 2-3.
Since $T = \sum N_i^e(x, y)T_i$ and all N_i^e are continuous across element boundaries, so is T, and an arbitrary mesh of 3-node triangular elements is conforming.



Pyramid function: global shape function pieced together from (flat) element shape functions. Note that the shape function, attached to node i, is defined everywhere but is nonzero only on elements surrounding node i.

3.1.2 4-Node Rectangular Element



Approximation: $T(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 + \alpha_4 xy$. Can construct shape functions by using $T(x_i, y_i) = T_i \Rightarrow$ 4 eqs / 4 unknowns $\Rightarrow T = \sum N_i T_i$.

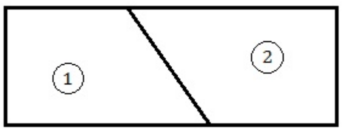
Alternatively, we can use repeated 1D interpolation (first in x, then in y):

$$\begin{aligned} T(x, y) &= N_{1x}T_{1y} + N_{2x}T_{2y} \\ &\quad \text{(keeping } y \text{ fixed; coefficients } T_{1y}, T_{2y} \text{ still depend on } y\text{)} \\ &= (1 - \frac{x}{L_1})T_{1y} + \frac{x}{L_1}T_{2y} \\ &= (1 - \frac{x}{L_1})[(1 - \frac{y}{L_2})T_1 + \frac{y}{L_2}T_4] + \frac{x}{L_1}[(1 - \frac{y}{L_2})T_2 + \frac{y}{L_2}T_3] \\ &= \sum N_i T_i, \end{aligned}$$

where

$$
\begin{aligned}
N_1 &= (1-\frac{x}{L_1})(1-\frac{y}{L_2}) = P_1^1(x)P_1^1(y)\\
N_2 &= \frac{x}{L_1}(1-\frac{y}{L_2}) = P_2^1(x)P_1^1(y)\\
N_3 &= \frac{x}{L_1}\frac{y}{L_2} = P_2^1(x)P_2^1(y)\\
N_4 &= (1-\frac{x}{L_1})\frac{y}{L_2} = P_1^1(x)P_2^1(y),
\end{aligned}
$$

where P_k^{n-1} are the Lagrange polynomials (2.2). The shape functions N_i are products of 2-node 1D shape functions and therefore products of Lagrange polynomials P_i^1 $(i = 1, 2)$. They are therefore called **Lagrange shape functions**. These elements are bilinear, i.e., linear along horizontal or vertical boundaries (where either $x =$ const or $y =$ const). So a mesh of such elements is conforming if sides are parallel to the x or y axes.



$$
\begin{aligned}
T^{①} &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy = a_1 + a_2 x + a_3 x^2\\
T^{②} &= \beta_1 + \beta_2 x + \beta_3 y + \beta_4 xy = b_1 + b_2 x + b_3 x^2
\end{aligned}
$$

However, if the common boundary of two neighbouring elements is not parallel to either the x or y axis, then the mesh is not conforming: T is given by a quadratic polynomial on this boundary and hence determined by three coefficients, but there are only two common nodes on the boundary which therefore do not uniquely fix T. T computed on the one element will therefore in general be different from T computed on the other element, implying nonconformity.

3.1.3 9-Node Rectangular Element

$$T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 y^2 + \alpha_6 xy + \alpha_7 x^2 y + \alpha_8 xy^2 + \alpha_9 x^2 y^2$$

Shape functions biquadratic, products of quadratic shape functions in both x and y, so again a Lagrange product element: $N_1(x, y) = P_1^2(x)P_1^2(y)$, etc.

3.1.4 8-Node Serendipity Element

We can construct shape functions the standard way: solving an 8×8 system of equations $T(x_i, y_i) = T_i$

We can however also construct them directly by using properties of shape functions.

Introduce normal (or natural) coordinates ξ, η. (The mapping between (x, y) and (ξ, η) is given by

$$
\begin{aligned}
x &= (x_1 + a) + \xi a = x_1 + (\xi + 1)a,\\
y &= y_1 + (\eta + 1)b,
\end{aligned}
$$

where $2a$ and $2b$ are the sides of the rectangle.)
Now consider a midside node, for instance 5. By the $N_i(\xi_j, \eta_j) = \delta_{ij}$ property of shape functions, we must have

$$N_5 = \alpha(1-\xi)(1+\xi)(1-\eta)$$

(zero along sides 2-6-3, 3-7-4 and 4-8-1). Here α is still an arbitrary parameter, fixed by $1 = N_5$(node 5)$= N_5(0, -1) = 2\alpha \Rightarrow \alpha = \frac{1}{2}$. Similarly for N_6, N_7, N_8.
Next consider a corner node, for instance 1. Again, by shape function properties, N_1 must have the factors $(1-\xi)$ and $(1+\xi)$. In addition, $N_1 = 0$ at nodes 5 and 8. It turns out that if we ask that $N_1 = 0$ along the entire line through nodes 5 and 8, we get the correct expression. Thus

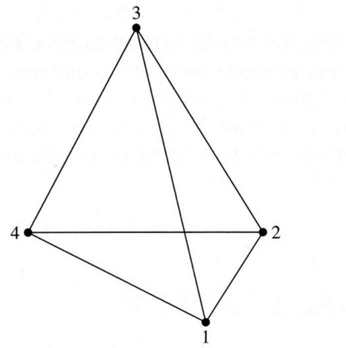
$$N_1 = \beta(1-\xi)(1-\eta)(1+\eta+\xi),$$

where β is still to be determined. It is fixed by $1 = N_1$(node 1)$= N_1(-1, -1) = -4\beta \Rightarrow \beta = -\frac{1}{4}$.
Similarly for N_2, N_3, N_4.
Thus we have

$$\begin{aligned}
N_1(\xi,\eta) &= -\frac{1}{4}(1-\xi)(1-\eta)(1+\xi+\eta), & N_2(\xi,\eta) &= -\tfrac{1}{4}(1+\xi)(1-\eta)(1-\xi+\eta),\\
N_3(\xi,\eta) &= -\frac{1}{4}(1+\xi)(1+\eta)(1-\xi-\eta), & N_4(\xi,\eta) &= -\tfrac{1}{4}(1-\xi)(1+\eta)(1+\xi-\eta),\\
N_5(\xi,\eta) &= \frac{1}{2}(1-\xi^2)(1-\eta), & N_6(\xi,\eta) &= \tfrac{1}{2}(1+\xi)(1-\eta^2),\\
N_7(\xi,\eta) &= \frac{1}{2}(1-\xi^2)(1+\eta), & N_8(\xi,\eta) &= \tfrac{1}{2}(1-\xi)(1-\eta^2).
\end{aligned}$$

3.2 3D Elements

3.2.1 4-Node Tetrahedral Element

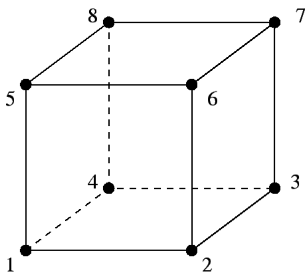


$$T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z$$

(linear in x, y, z)

3.2.2 8-Node Prism (or Brick) Element

Shape functions are products of linear Lagrange polynomials in x, y, z:



$$N_1 = P_1^1(x)P_1^1(y)P_1^1(z), \ldots, \text{ etc.}$$

3.3 2D Example

3.3.1 Torsion of a Square Shaft

Governing equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + 2G\theta = 0 \qquad \text{(Poisson equation)}$$

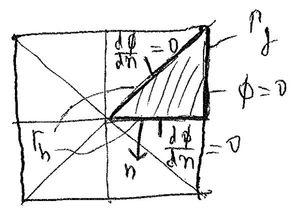
ϕ is the Prandtl stress function, G is the shear modulus and θ is the twist rate. Note that this torsion problem is a pure shear problem; the only nonzero stresses are the shear stresses σ_{xz} and σ_{yz}, which can be obtained as $\sigma_{xz} = \frac{\partial \phi}{\partial y}, \sigma_{yz} = -\frac{\partial \phi}{\partial x}$.

Derivation of the boundary condition on ϕ:

The gradient of the stress function ϕ is $\nabla\phi = (\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y})$. The stress resultant vector $v = (\sigma_{xz}, \sigma_{yz}) = (\frac{\partial \phi}{\partial y}, -\frac{\partial \phi}{\partial x})$ is therefore always perpendicular to the gradient: $v \cdot \nabla\phi = 0$.

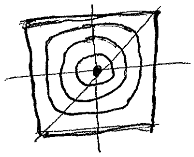
The gradient is perpendicular to the contour lines of ϕ, i.e., the constant-ϕ lines (and pointing in the direction of increasing ϕ), so v is tangential to the contour lines (one says that contour lines are lines of shear stress). Now, since there is no component of shear stress normal to the boundary (only tangent to the boundary), the outer boundary is a line of shear stress and hence also a contour for ϕ. We can choose $\phi = 0$ (only potential differences matter – we always take derivatives of ϕ), which gives us the boundary condition: $\phi = 0$ on Γ.
By symmetry we need to consider only an eighth of the cross-section. On this triangular domain the problem becomes

cross-section:



$$
\begin{cases}
\dfrac{\partial^2\phi}{\partial x^2} + \dfrac{\partial^2\phi}{\partial y^2} + 2G\theta = 0 \\
\phi = 0 \text{ on } \Gamma_g \\
\dfrac{\mathrm{d}\phi}{\mathrm{d}n} = 0 \text{ on } \Gamma_h \qquad (\dfrac{\mathrm{d}\phi}{\mathrm{d}n} = \nabla\phi \cdot n)
\end{cases}
$$

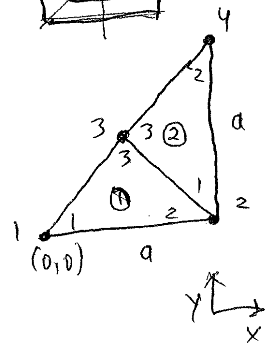
potential lines normal to Γ_h (3 axes of symmetry):



We model the problem by a mesh of two three-node triangular elements.
Shape functions:

$$N_1^e = \frac{1}{2A}[x_2y_3 - x_3y_2 + (y_2 - y_3)x + (x_3 - x_2)y]$$

...



Weak formulation (v arbitrary test function):

$$
\begin{aligned}
0 &= \iint (\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + 2G\theta)v\mathrm{d}x\mathrm{d}y \\
&\underset{\overline{\overline{\text{Gauss-Green}}}}{} \iint (-\frac{\partial\phi}{\partial x}\frac{\partial v}{\partial x} - \frac{\partial\phi}{\partial y}\frac{\partial v}{\partial y} + 2G\theta v)\mathrm{d}x\mathrm{d}y + \int_\Gamma (\frac{\partial\phi}{\partial x}n_x + \frac{\partial\phi}{\partial y}n_y)v\mathrm{d}s.
\end{aligned}
$$

By the gradient boundary condition we can restrict the boundary integral to the Γ_g part of the boundary. We then have the remaining (Dirichlet) boundary condition

$$\phi = 0 \quad \text{on} \quad \Gamma_g.$$

Interpolation: $\phi = \sum_{j=1}^{n} N_j \phi_j$ (in terms of global shape functions N_j).
Galerkin: $v_i = N_i$ $(i = 1, \ldots, n)$.
$\Rightarrow$ FE eq.: $\sum_{j=1}^{n} K_{ij}\phi_j = f_{li} + f_{bi}$ $(Ku = f_l + f_b)$.
Stiffness matrix:

$$K_{ij} = \iint (\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y})\mathrm{d}x\mathrm{d}y.$$

Load vector:

$$f_{li} = 2G\theta \iint N_i \mathrm{d}x\mathrm{d}y.$$

Boundary vector:

$$f_{b_i} = \int_{\Gamma_g} N_i \frac{\mathrm{d}\phi}{\mathrm{d}n}\mathrm{d}s.$$

We compute data element by element using the element shape functions N_i^e.
Element ①:

$$K = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

Load vector: $f_{l_i} = 2G\theta \iint N_i^e \mathrm{d}x\mathrm{d}y$.
Recall that $N_i^e = \eta_i$ (area coordinates), so we can use the integration formula

$$\iint \eta_1^n \eta_2^m \eta_3^p \mathrm{d}x\mathrm{d}y = \frac{m!n!p!}{(m+n+p+2)!}(2A)$$

to get

$$f_l = \frac{2}{3}G\theta A \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

The boundary vector is best added after assembly so that it does not have to be computed for each element individually.

Element ②:
Note that because the problem is uniform (no explicit x or y dependence in data) and because the two elements are congruent and we use the same local node numberings, we have $K^2 = K^1$, $f_l^2 = f_l^1$.

Assembly:
Element ①

$$u^1 = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} = A^1 u.$$

Element ②:

$$u^2 = \begin{pmatrix} \phi_2 \\ \phi_4 \\ \phi_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} = A^2 u$$

(in terms of the global nodal degrees of freedom u).
Then on element ① we can write

$$\begin{aligned} K^1 u^1 = f^1 \qquad & \Rightarrow \quad K^1 A^1 u = f^1 \\ & \Rightarrow \quad A^{1T} K^1 A^1 u = A^{1T} f^1. \end{aligned}$$

Similar for element ②.
Then sum over elements:

$$\sum_{e=1}^{2} A^{eT} K^e A^e u = \sum_{e=1}^{2} A^{eT} f_l^e \quad \text{or} \quad Ku = f_l,$$

where $K = \sum_e A^{eT} K^e A^e$ and $f_l = \sum_e A^{eT} f_l^e$.
The global stiffness matrix is then

$$K = \frac{1}{2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 2 & -2 & 0 \\ -1 & -2 & 4 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

and the global load vector

$$f_l = \frac{2}{3} G\theta A \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}.$$

Global boundary vector (in terms of global shape functions):

$$f_{b_i} = \int_{\Gamma_g} N_i \frac{\mathrm{d}\phi}{\mathrm{d}n} \mathrm{d}s.$$

Along Γ_g the only nonzero global shape functions are N_2 and N_4, so we can write

$$f_b = \begin{pmatrix} 0 \\ R_2 \\ 0 \\ R_4, \end{pmatrix}, \qquad R_i = \int_{\Gamma_g} N_i \frac{\mathrm{d}\phi}{\mathrm{d}n} \mathrm{d}s.$$

We thus have the FE equation

$$\frac{1}{2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 2 & -2 & 0 \\ -1 & -2 & 4 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 = 0 \\ \phi_3 \\ \phi_4 = 0 \end{pmatrix} = \frac{2}{3} G\theta A \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ R_2 \\ 0 \\ R_4 \end{pmatrix},$$

where the Dirichlet boundary conditions ($\phi = 0$ on Γ_g, hence $\phi_2 = 0$ and $\phi_4 = 0$) have been applied.

Solution:
Solve for ϕ_1 and ϕ_3 from 1st and 3rd eq. and then (if required) solve for R_2 and

R_4 from 2nd and 4th eq.

$$\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 4 \end{pmatrix}\begin{pmatrix} \phi_1 \\ \phi_3 \end{pmatrix} = \frac{2}{3}G\theta A\begin{pmatrix} 1 \\ 2 \end{pmatrix} \Rightarrow \begin{cases} \phi_3 = \dfrac{4}{3}G\theta A \\ \phi_1 = \dfrac{8}{3}G\theta A \end{cases}$$

$$R_2 = -\frac{8}{3}G\theta A, \quad R_4 = -\frac{4}{3}G\theta A.$$

Three checks to see if the solution makes sense:

- ϕ at the centre of the shaft $= \phi(0,0) = \phi_1$.
  Thus FE gives $\frac{\phi_1}{2G\theta} = \frac{4}{3}A \overset{A=\frac{a^2}{4}}{=\!=\!=} \frac{1}{3}a^2$.
  Exact solution (series expansion): $\frac{\phi(0,0)}{2G\theta} \simeq 0.2947a^2$ (13 % error).

- Torsional stiffness $C = \frac{M}{\theta}$, where M is twisting moment, which may be obtained as

  $$M = 2\iint \phi \,\mathrm{d}x\mathrm{d}y$$

  Thus we compute

  $$\begin{aligned} M =& 2\left(8\iint_{1} \phi \,\mathrm{d}x\mathrm{d}y + 8\iint_{2} \phi \,\mathrm{d}x\mathrm{d}y\right) \\ =& 16\iint_{1} (N_1^1\phi_1 + N_2^1\phi_2 + N_3^1\phi_3)\,\mathrm{d}x\mathrm{d}y + 16\iint_{2} (N_1^2\phi_2 + N_2^2\phi_4 + N_3^2\phi_3)\,\mathrm{d}x\mathrm{d}y \\ =& \cdots = \frac{256}{9}G\theta A^2 \quad \left(\text{since } \textstyle\iint N_i^e \,\mathrm{d}x\mathrm{d}y = \frac{A}{3} \text{ by the integration formula (3.1)}\right). \end{aligned}$$

  Hence $C = \frac{M}{\theta} = \frac{256}{9}GA^2 = \frac{1}{9}G\bar{A}^2 \simeq 0.1111G\bar{A}^2$, where $\bar{A} = 16A$ is the cross-sectional area of the shaft.
  Exact value: $C = k_1 G\bar{A}^2$ with $k_1 \simeq 0.1406$ for a square shaft (21 % error).

- Contour curves.
  We need ϕ over the full domain. Computing element by element:

  Element 1:

  $$\begin{aligned} \phi &= N_1\phi_1 + N_2\phi_2 + N_3\phi_3 \quad \text{(globally)} \\ &= N_1^1\phi_1 + N_2^1\phi_2 + N_3^1\phi_3 \quad \text{(element)} \\ &= \frac{8}{3}G\theta A(1 - \frac{x}{a}). \end{aligned}$$

  Element 2:

  $$\begin{aligned} \phi &= N_2\phi_2 + N_3\phi_3 + N_4\phi_4 \quad \text{(globally)} \\ &= N_1^2\phi_2 + N_3^2\phi_3 + N_2^2\phi_4 \quad \text{(element)} \\ &= \frac{8}{3}G\theta A(1 - \frac{x}{a}). \end{aligned}$$

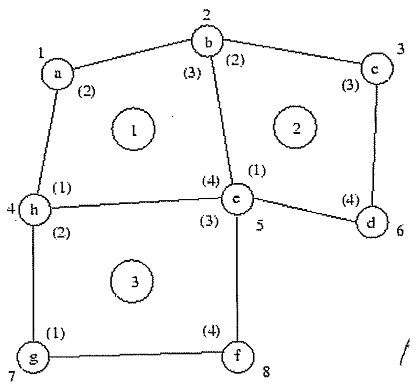
  Conclusion: $\phi|_{\text{element1}} = \phi|_{\text{element2}}$.
  Contour curve: $\phi =$const $\Rightarrow$ $x =$ const (i.e., vertical lines).
  This makes good sense to this (linear) order of approximation.

Exercise:
Compute the torsional stiffness with a mesh of 2 4-node rectangular elements for a quarter of the cross-section.

3.3.2 Assembly Using the Connectivity Matrix

To assemble the element data into global data, in the above example we used the A^e matrices containing only 0s and 1s. On a large mesh you would mostly be storing zeroes to keep the A^e in memory, and when computing, for instance, the global stiffness matrix you would mostly be multiplying by zero $\Rightarrow$ very inefficient.
An alternative way is as follows.
Note that what the A^e matrices do is put the 3×3 element matrices in the right position in the global matrix K:
element ① goes to the 1-2-3 positions, element ② goes to the 2-4-3 positions.
This network topology is conveniently encoded in a matrix, or table, called the **connectivity matrix**.
This matrix lists for each element the global node numbers in the order of the element node numbers (1, 2, 3, 4 for a 4-node element). Since the element shape functions are always constructed in terms of the element node numbering, each row of the connectivity matrix then specifies which position the corresponding element data K^e_{ij} should go in the global matrix K_{ij}. Using this connectivity matrix we do not have to multiply any matrices; we put the element data directly in the right place in the global matrix.

Example:



$$C = \begin{pmatrix} 4 & 1 & 2 & 5 \\ 5 & 2 & 3 & 6 \\ 7 & 4 & 5 & 8 \end{pmatrix}$$

global nodes in order of element nodes

$$\begin{array}{ll} K^1_{11} \to K_{44} & \text{slot} \\ K^1_{12} \to K_{41} & \text{slot} \\ K^2_{32} \to K_{32} & \text{etc.} \end{array}$$

Assembly algorithm:

initialise K to the zero matrix
for $e = 1, \ldots, N$ (N # of elements)
  for $i = 1, \ldots, n$ (n # of nodes per element)
    for $j = i, \ldots, n$ (by symmetry only $j \geq i$ required)
      find global nodes p and q corresponding to element nodes i and j
      (from connectivity matrix)
      compute, or retrieve, K^e_{ij} and add the result to K_{pq}

(and also to K_{qp} if $q \neq p$)

3.4 Bandwidth

The global stiffness matrix K depends on the choice of global node numbering (although the final solution of course doesn't). For instance, in the torsion problem we had $K_{14} = 0$ because there was no element containing both global nodes 1 and 4.
Let $R = \max_e\{\,|i-j|\,|\ i \text{ and } j \text{ are global nodes in element } e\}$.
Then $K_{ij} = 0$ if $|i-j| > R$.
$B \overset{\text{def.}}{=\!=} R+1$ is called the **bandwidth** of the global matrix K.
Codes (e.g., Matlab) can take advantage of low bandwidth both in storage (sparse matrices) and in computational efficiency (speed).
For instance, when using Gaussian elimination to solve $Ku = f$, the computational cost (i.e., the number of operations, that is, multiplications and additions) scales as B^2 (or, more precisely, as $B^2 n$ as the computational cost of solving an $n \times n$ system without any structure scales as n^3). For large problems it is therefore important to choose a global node numbering such that B is as low as possible.

3.5 Accuracy and Efficiency

FE approximation for 1D problems: $T = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \ldots$
Taylor expansion of the exact solution T gives

$$\begin{aligned} T = T(x_i) &+ \frac{\mathrm{d}T}{\mathrm{d}x}|_{x=x_i}(x-x_i) + \frac{1}{2}\frac{\mathrm{d}^2T}{\mathrm{d}x^2}|_{x=x_i}(x-x_i)^2 \\ &+ \cdots + \frac{1}{n!}\frac{\mathrm{d}^nT}{\mathrm{d}x^n}|_{x=x_i}(x-x_i)^n + \frac{1}{(n+1)!}\frac{\mathrm{d}^{n+1}T}{\mathrm{d}x^{n+1}}|_{x=\xi}(x-x_i)^{n+1} \\ &\text{for some } \xi \in [x_i, x_j]. \end{aligned}$$

Thus, truncation after a finite number of terms gives an error of $O(h^{p+1})$ ('of order h^{p+1}'), where h is the size of the element and p is the highest full order in the expansion of T. $O(h^{p+1})$ is merely the truncation error in the interpolation (the coefficients α_i, or the nodal values T_i, are generally computed with their own errors). However, for linear problems no significant additional error occurs and the error estimate holds for the full solution.
This error analysis shows in particular that if no constant term, α_1, is included in the approximation, then the error is $O(1)$, i.e., it doesn't decrease with element size h, and therefore mesh refinement is pointless for accuracy purposes. The need for a constant term for convergence (since in the limit $h \to 0$ any continuous solution tends to a constant on an element) is called the **consistency** requirement of finite elements.

2D approximation: $T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + \ldots$.

Taylor expansion:

$$T = T(x_i, y_i) + \frac{\partial T}{\partial x}|_i (x - x_i) + \frac{\partial T}{\partial y}|_i (y - y_i)$$
$$+ \frac{1}{2}\frac{\partial^2 T}{\partial x^2}|_i (x - x_i)^2 + \frac{1}{2}\frac{\partial^2 T}{\partial y^2}(y - y_i)^2 + \frac{\partial^2 T}{\partial x \partial y}|_i (x - x_i)(y - y_i) + \dots$$

Again we find an error of the form $O(h^{p+1})$. (This estimate assumes a 'balanced mesh' in which both x and y dimensions of all elements are of order h.)

For instance, for triangular elements:
3-node: $T = \alpha_1 + \alpha_2 x + \alpha_3 y \rightarrow$ error $O(h^2)$
6-node: $T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 y^2 + \alpha_6 xy \rightarrow$ error $O(h^3)$
10-node: error $O(h^4)$ (all cubic terms included, but not all quartic terms, hence $p = 3$)
These elements are called **order-complete** because all terms are included up to a certain order and not more (note that this is governed by Pascal's triangle). Such elements are numerically efficient as they have the lowest number of degrees of freedom required to obtain a certain degree of accuracy. The 9-node triangular element is only $O(h^3)$ and is not order-complete.

For rectangular elements:
4-node: $T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy$
$\rightarrow$ error $O(h^2)$ because not all quadratic terms are included.
9-node biquadratic: $T = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + \alpha_5 x^2 + \alpha_6 y^2 + \alpha_7 x^2 y + \alpha_8 xy^2 + \alpha_9 x^2 y^2$
$\rightarrow$ error $O(h^3)$ because not all cubic terms are included.
In this case the central node is useless:

- it doesn't increase accuracy, as the error is still $O(h^3)$,
- it doesn't help with conformity because it is not on the boundary.

It only increases the dimension of the problem and hence the computational cost. Such a node is called a **parasitic node**. We already saw in Section 3.1 that instead of this 9-node element the 8-node serendipity element, without the central node, is frequently used.

The error $O(h^{p+1})$ suggests two ways of increasing accuracy of the FE solution:

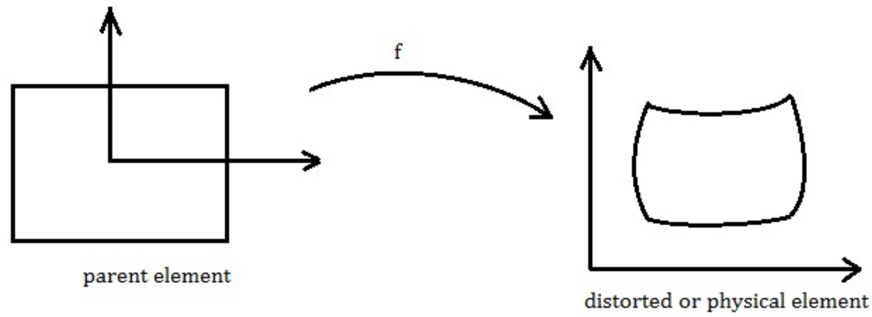
- reduce h by refining the mesh (more elements) (the ***h*-method**),
- increase p by using higher-order shape functions (more nodes per element) (the ***p*-method**).

Chapter 4

Isoparametric Elements

Away from curved boundaries either triangular or rectangular elements work fine. Near a curved boundary one could then refine the mesh to model the boundary more accurately, but this increases the number of degrees of freedom and therefore comes at a computational cost. It would be nice to have elements of more flexible shape that can map a curved boundary more accurately. This flexibility is offered by isoparametric elements.

The idea is to use a mapping from a regular *parent element* to a *distorted*



or *physical element*. We specify shape functions on the parent element. Let (x, y) be the coordinates for the physical element and (ξ, η) the coordinates for the parent element. For the mapping, f, we can then write: $f : (\xi, \eta) \mapsto (x(\xi, \eta), y(\xi, \eta))$. If the mapping (or transformation) is sufficiently "nice", then properties of the parent element (e.g., conformity) will carry over to the distorted element. In particular we require:

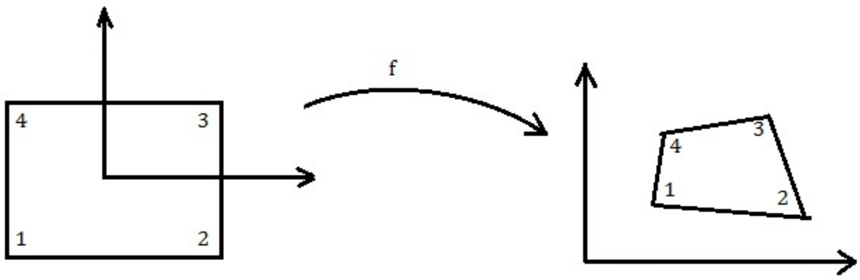
- f continuous
- f invertible (so that we can move back and forth between distorted and parent element). In particular, this means that the Jacobian matrix of partial derivatives
$$J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix}$$
must be regular, i.e, have $\det J \neq 0$.

It turns out we can use the shape functions to construct a suitable map:

$$x = x(\xi, \eta) = \sum_{i=1}^{n} N_i(\xi, \eta) x_i,$$
$$y = y(\xi, \eta) = \sum_{i=1}^{n} N_i(\xi, \eta) y_i,$$

where (x_i, y_i) are the coordinates of the nodes of the physical element. (The element is called isoparametric because the same 'parameters', i.e., shape functions, are used for both interpolation and transformation.)

4.1 4-Node Isoparametric Quadrilateral Element



Shape functions on (ξ, η) known:

$$N_1^e = \tfrac{1}{4}(\xi - 1)(\eta - 1)$$
$$N_2^e = -\tfrac{1}{4}(\xi + 1)(\eta - 1)$$
$$N_3^e = \tfrac{1}{4}(\xi + 1)(\eta + 1)$$
$$N_4^e = -\tfrac{1}{4}(\xi - 1)(\eta + 1)$$
$$x = N_1(\xi,\eta)x_1 + N_2(\xi,\eta)x_2 + N_3(\xi,\eta)x_3 + N_4(\xi,\eta)x_4$$
$$y = N_1(\xi,\eta)y_1 + N_2(\xi,\eta)y_2 + N_3(\xi,\eta)y_3 + N_4(\xi,\eta)y_4$$

By the properties of N_i, corner nodes are mapped into corner nodes:

$$x(-1,-1) = x_1, \quad x(1,-1) = x_2, \quad x(1,1) = x_3, \quad x(-1,1) = x_4,$$
$$y(-1,-1) = y_1, \quad y(1,-1) = y_2, \quad y(1,1) = y_3, \quad y(-1,1) = y_4.$$

Now consider a side, $\xi = -1$:

$$x(-1, \eta) = -\frac{1}{2}(\eta - 1)x_1 + \frac{1}{2}(\eta + 1)x_4,$$
$$y(-1, \eta) = -\frac{1}{2}(\eta - 1)y_1 + \frac{1}{2}(\eta + 1)y_4.$$

By eliminating η we find a linear relation between x and y:

$$ax + by = 1.$$

Thus, side 14 maps into a straight line connecting nodes 1 and 4. The same is true for the other sides.
Note though that a diagonal line is not mapped into a straight line, because the N_i are bilinear but not linear (but linear if ξ =const, or η =const).
This isoparametric element is conforming because the 4-node square element is and the map f is continuous, so preserve continuity.
Note that this isoparametric element also solves the problem of quadrilateral elements with sides not parallel to coordinate axes being nonconforming.

What about the invertibility of the Jacobian?

$$J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix}$$
$$\Rightarrow \det J = c_1 + c_2\xi + c_3\eta,$$

for some constants c_i.
So $\det J$ is a linear function of ξ and η (the $\xi\eta$ term cancels). This means that if $\det J > 0$ at all four nodes then it must be > 0 on the entire element.
Now, at node 1, $(\xi, \eta) = (-1, -1)$, we compute

$$J = \begin{pmatrix} \frac{1}{2}(x_2 - x_1) & \frac{1}{2}(x_4 - x_1) \\ \frac{1}{2}(y_2 - y_1) & \frac{1}{2}(y_4 - y_1) \end{pmatrix}$$
$$\det J = \frac{1}{4}(x_2 - x_1)\frac{1}{2}(y_4 - y_1) - \frac{1}{2}(x_4 - x_1)\frac{1}{2}(y_2 - y_1)$$
$$= \frac{1}{4} l_1 \times l_2 = \frac{1}{4}|l_1||l_2|\sin\theta \quad (\times = \text{cross product})$$
$$l_1 = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}, \quad l_2 = \begin{pmatrix} x_4 - x_1 \\ y_4 - y_1 \end{pmatrix}$$

Similar for the other three nodes.
Conclusion: $\det J$ is positive at all four nodes provided the enclosed angles θ are less than π, or:
$\det J$ is non-vanishing on the entire element if and only if the quadrilateral element is convex, i.e., all its angles are less than π.
For highly distorted elements with $\theta > \pi$, $\det J$ will vanish on the element and we cannot move back anf forth between parent and physical element as this will involve the inverse of J (see later), which does not exist.

4.2 8-Node Isoparametric Quadrilateral Element

Again,

$$\text{nodes} \rightarrow \text{nodes}$$
$$\text{sides} \rightarrow \text{curved sides}$$

Sides of the physical element are curved (in fact quadratic) because the N_i are cubic, and quadratic for ξ =const or η =const.
We can extend this idea: for a boundary described by a higher-degree polynomial we can take more nodes in the parent so that the shape functions have the required degree on the boundary.

4.3 Transformation of the Load Vector

Let the load vector be given in the form

$$f_{li} = \iint_\Omega N_i(\xi,\eta)Q(x,y)\mathrm{d}x\mathrm{d}y$$

To see how this transforms to (ξ,η) coordinates we first recall from calculus how integrals transform under a change of variables:
In 1D:

$$\int f(x)\mathrm{d}x \xlongequal{x=g(y)} \int f(g(y))g\prime(y)\mathrm{d}y.$$

In 2D:

$$\iint_\Omega f(x,y)\mathrm{d}x\mathrm{d}y \xlongequal{F} \iint_\Omega f(x(u,v),y(u,v))|\det J|\mathrm{d}u\mathrm{d}v$$
$$\begin{pmatrix} x \\ y \end{pmatrix} = F\begin{pmatrix} u \\ v \end{pmatrix}, \quad J = \frac{\partial(x,y)}{\partial(u,v)} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix}.$$

Load vector:

$$f_{l_i} = \iint_\Omega N_i(\xi,\eta)Q(x,y)\mathrm{d}x\mathrm{d}y = \int_{-1}^{1}\int_{-1}^{1} N_i(\xi,\eta)Q(x(\xi,\eta),y(\xi,\eta))|\det J|\mathrm{d}\xi\mathrm{d}\eta.$$

4.4 Transformation of the Boundary Vector

$$f_b = \int_\Gamma f(x,y)\mathrm{d}s = \int_{\Gamma_1}\dots + \int_{\Gamma_2}\dots + \int_{\Gamma_3}\dots + \int_{\Gamma_4}\dots, \qquad \mathrm{d}s = \sqrt{(\mathrm{d}x)^2+(\mathrm{d}y)^2},$$
$$\text{where (chain rule)} \quad \mathrm{d}x = \frac{\partial x}{\partial \xi}\mathrm{d}\xi + \frac{\partial x}{\partial \eta}\mathrm{d}\eta, \quad \mathrm{d}y = \frac{\partial y}{\partial \xi}\mathrm{d}\xi + \frac{\partial y}{\partial \eta}\mathrm{d}\eta.$$

Along Γ_1 and Γ_3, we have $\mathrm{d}\eta = 0$.
Along Γ_2 and Γ_4, we have $\mathrm{d}\xi = 0$.
(Note: along a diagonal side, as in a triangular element, you would have, e.g., $\mathrm{d}\xi = -\mathrm{d}\eta$.)
Thus

$$\int_{\Gamma_1} f(x,y)\mathrm{d}s = \int_{-1}^{1} f(x(\xi,\eta),y(\xi,\eta))\sqrt{(\frac{\partial x}{\partial \xi})^2 + (\frac{\partial y}{\partial \xi})^2}\mathrm{d}\xi \quad \text{with } \eta = -1,$$
$$\int_{\Gamma_3} f(x,y)\mathrm{d}s = \int_{1}^{-1} f(x(\xi,\eta),y(\xi,\eta))\sqrt{(\frac{\partial x}{\partial \xi})^2 + (\frac{\partial y}{\partial \xi})^2}\mathrm{d}\xi \quad \text{with } \eta = 1.$$

Similarly,

$$\int_{\Gamma_2} f(x,y)\mathrm{d}s = \int_{-1}^{1} f(x(\xi,\eta),y(\xi,\eta))\sqrt{(\frac{\partial x}{\partial \eta})^2 + (\frac{\partial y}{\partial \eta})^2}\mathrm{d}\eta \quad \text{with } \xi = 1,$$

$$\int_{\Gamma_4} f(x,y)\mathrm{d}s = \int_{1}^{-1} f(x(\xi,\eta),y(\xi,\eta))\sqrt{(\frac{\partial x}{\partial \eta})^2 + (\frac{\partial y}{\partial \eta})^2}\mathrm{d}\eta \quad \text{with } \xi = -1.$$

4.5 Transformation of the Stiffness Matrix

Stiffness matrix:

$$K_{ij} = \iint (\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y})\mathrm{d}x\mathrm{d}y \quad \text{(e.g., Poisson equation)}.$$

More generally, we might consider the equation

$$a_{11}\frac{\partial^2 \phi}{\partial x^2} + 2a_{12}\frac{\partial^2 \phi}{\partial x \partial y} + a_{22}\frac{\partial^2 \phi}{\partial y^2} + c = 0.$$

Via the weak formulation we then find the stiffness matrix in the form

$$K_{ij} = \iint (\frac{\partial N_i}{\partial x} \quad \frac{\partial N_i}{\partial y}) D \begin{pmatrix} \frac{\partial N_j}{\partial x} \\ \frac{\partial N_j}{\partial y} \end{pmatrix} \mathrm{d}x\mathrm{d}y, \quad \text{where } D = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}.$$

Note that $N_i = N_i(\xi, \eta)$, so we need to transform $\frac{\mathrm{d}}{\mathrm{d}x}, \frac{\mathrm{d}}{\mathrm{d}y}$ to $\frac{\mathrm{d}}{\mathrm{d}\xi}, \frac{\mathrm{d}}{\mathrm{d}\eta}$. To this end, we compute (chain rule)

$$\begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} = J^T \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix}.$$

Now take the inverse

$$\begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{pmatrix}.$$

Then

$$K_{ij} = \int_{-1}^{1}\int_{-1}^{1} (\frac{\partial N_i}{\partial \xi} \quad \frac{\partial N_i}{\partial \eta}) J^{-1} D (J^T)^{-1} \begin{pmatrix} \frac{\partial N_j}{\partial \xi} \\ \frac{\partial N_j}{\partial \eta} \end{pmatrix} |\det J| \mathrm{d}\xi\mathrm{d}\eta.$$

Note that $(J^T)^{-1} = (J^{-1})^T$.

Chapter 5

Fourth-order problems: The Euler-Bernoulli Beam

Two main assumptions:

- neglect shear deformation (normal sections remain normal under deformation – **Bernoulli hypothesis**)
- neglect rotational inertia

Force balance in the vertical (lateral) direction (including inertia):

$$(S + \mathrm{d}S)\cos\theta - S\cos\theta + f\mathrm{d}x = \rho A \mathrm{d}x \frac{\partial^2 v}{\partial t^2}$$

v lateral deflection
f distributed load (e.g., gravity)
S shear force
M bending moment
ρ mass density (volumetric)
A cross-sectional area
We assume small deflections $\theta << 1 \Rightarrow \cos\theta \approx 1$ so that vertical force balance gives (divide by $\mathrm{d}x$ and take the limit $\mathrm{d}x \to 0$):

$$\frac{\partial S}{\partial x} + f = \rho A \frac{\partial^2 v}{\partial t^2}.$$

Moment balance (about the centre of gravity):

$$M + \mathrm{d}M - M + S\frac{\mathrm{d}x}{2} + (S + \mathrm{d}S)\frac{\mathrm{d}x}{2} = 0 \quad \rightarrow \quad \frac{\mathrm{d}M}{\mathrm{d}x} + S = 0$$

(right-hand side zero because of assumption of no rotational inertia).
Constitutive relation (linear relationship between bending moment M and curvature κ):

$$M = EI\kappa, \qquad \kappa \approx \frac{\mathrm{d}^2 v}{\mathrm{d}x^2} \text{ is the (approximate) curvature.}$$

E Young's modulus
I second moment of area of the cross-section.
Thus we obtain the **Euler-Bernoulli equation**

$$-\frac{\partial^2}{\partial x^2}(EI(x)\frac{\partial^2 v}{\partial x^2}) + f(x,t) = \rho(x)A(x)\frac{\partial^2 v}{\partial t^2}.$$

Statics case (writing straight d's):

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}(EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}) = f.$$

This is a 4th-order equation requiring four boundary conditions. Possible BCs:

- Simply-supported (pinned) at both $x = 0$ and $x = L$:
  $v(0) = v(L) = 0$
  $M(0) = M(L) = 0 \quad \Rightarrow \quad v''(0) = v''(L) = 0$
- Cantilever (fixed at $x = 0$, free at $x = L$):
  $v(0) = 0, \quad \theta(0) = \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0$
  $M(L) = 0 \quad \Rightarrow \quad v''(L) = 0$
  $S(L) = 0 \quad \Rightarrow \quad v'''(0) = 0$

Weak formulation (arbitrary test function w):

$$\begin{aligned} 0 &= \int_0^L [\frac{\mathrm{d}^2}{\mathrm{d}x^2}(EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}) - f]w\mathrm{d}x \\ &\overset{\text{ibp }(2\times)}{=\!=\!=} \int_0^L [EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} - fw]\mathrm{d}x - [EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\frac{\mathrm{d}w}{\mathrm{d}x}]_0^L + [\frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2})w]_0^L, \end{aligned}$$

or

$$\int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\mathrm{d}x = \int_0^L fw\,\mathrm{d}x + \left[M\frac{\mathrm{d}w}{\mathrm{d}x}\right]_0^L + [Sw]_0^L\,. \tag{5.1}$$

For this to make sense, we require for any trial solution $\tilde{v}$ that $\frac{\mathrm{d}\tilde{v}}{\mathrm{d}x}$ be continuous and piecewise differentiable (compare with the 2nd-order case, where $\tilde{v}$, i.e., the field itself, had to be piecewise differentiable).
Idea: interpolate not only the displacement v but also the slope $\theta = \frac{\mathrm{d}v}{\mathrm{d}x}$, i.e., two degrees of freedom per node.
Thus approximation (for 2-node beam element): $v = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$.
Consistency: $v(0) = u_1$, $\frac{\mathrm{d}v}{\mathrm{d}x}(0) = u_2$, $v(L) = u_3$, $\frac{\mathrm{d}v}{\mathrm{d}x}(L) = u_4$.
We can then write $v(x) = \sum_{i=1}^4 N_i(x)u_i$, with element shape functions

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

Note that instead of the δ-function property of shape functions for 2nd-order equations we now have the (less useful) relations

$$\begin{aligned}
&N^e_{2i-1}(x_j) = \delta_{ij}, \\
&\frac{\mathrm{d}N^e_{2i}}{\mathrm{d}x}(x_j) = \delta_{ij} \quad (i,j = 1,2).
\end{aligned}$$

Polynomials that interpolate both function values and derivatives are called **Hermite polynomials**.
The global shape functions based on these Hermite polynomials are often referred to as C^1 shape functions, where C^k indicates the differentiability class of functions (a C^k function is k times continuously differentiable, here $k = 1$). Similarly, Lagrange shape functions are also called C^0 shape functions.

Hermite interpolation (4th-order problem)	Lagrange interpolation (2nd-order problem)
C^1 polynomials	C^0 polynomials
continuous, continuous 1st derivative	continuous

Interpolation: $v = \sum_{j=1}^{4} N_j u_j$.
Galerkin: $w = N_i \quad (i = 1,2,3,4)$.
FE equation (after substitution into (5.1)): $Ku = f_l + f_b$,
where

$$\begin{aligned}
K_{ij} &= \int_0^L \frac{\mathrm{d}^2 N_i}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 N_j}{\mathrm{d}x^2} \mathrm{d}x \qquad \text{(stiffness matrix)}, \\
f_{l_i} &= \int_0^L f N_i \mathrm{d}x \qquad \text{(load vector)}, \\
f_{b_i} &= [SN_i]_0^L + [M \frac{\mathrm{d}N_i}{\mathrm{d}x}]_0^L \qquad \text{(boundary vector)}.
\end{aligned}$$

Evaluation of element data (for a uniform beam, i.e., E, I, f constant):

$$K = \frac{EI}{L^3} \begin{pmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{pmatrix},$$

$$f_l = \frac{fL}{12} \begin{pmatrix} 6 \\ L \\ 6 \\ -L \end{pmatrix}, \qquad f_b = \begin{pmatrix} -S(0) \\ -M(0) \\ S(L) \\ M(L) \end{pmatrix}.$$

Point loads, as a special case of (very concentrated) distributed loads:

- For point force at $x = a$, $f = P\delta(x-a)$:

  $$f_{l_i} = \int_0^T f N_i \mathrm{d}x = P \int_0^T N_i \delta(x-a) \mathrm{d}x = PN_i(a) \quad \to \quad f_l = \begin{pmatrix} PN_1(a) \\ PN_2(a) \\ PN_3(a) \\ PN_4(a) \end{pmatrix}.$$

  So in general a point force induces both nodal forces and nodal moments. However, if the load is applied at a node, say $x = 0$, then (by shape function properties)

$$f_l = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

- Point moment M at $x = a$ is equivalent to a couple of forces (as $b \to 0$):

$$\begin{aligned} f_{li} &= -PN_i(a) + PN_i(a+b) \\ &\overset{\text{Taylor}}{=\!=\!=} -PN_i(a) + PN_i(a) + P\frac{\mathrm{d}N_i}{\mathrm{d}x}(a)b + O(b^2) \\ f_l &= M \begin{pmatrix} \frac{\mathrm{d}N_1}{\mathrm{d}x}(a) \\ \cdot \\ \cdot \\ \frac{\mathrm{d}N_4}{\mathrm{d}x}(a) \end{pmatrix} \qquad (M = Pb). \end{aligned}$$

If applied at node $x = 0$, then

$$f_l = \begin{pmatrix} 0 \\ M \\ 0 \\ 0 \end{pmatrix}.$$

We see that loads applied at nodes give contributions only at those nodes. Such loads can be treated as boundary conditions and applied to the boundary vector after assembly (in order to avoid having to divide the load P or M over two elements and then having to put element contributions together again).

5.0.1 Beam example

We consider a uniform Euler-Bernoulli beam of length L that is fixed at $x = 0$, supported by an end spring of stiffness k at $x = L$ and subjected to a uniform distributed load $-f$ acting (downwards) over the left half of the beam ($0 \le x \le L/2$) and a point load $-Q$ at $x = L/2$. We model this problem using a mesh of two 2-node beam elements of length $l = L/2$.
Boundary conditions: $v(0) = 0 = \theta(0)$, $S(L) = -kv(L)$, $M(L) = 0$.
FE equation (with all boundary conditions applied):

$$\frac{EI}{l^3}\begin{pmatrix} 12 & 6l & -12 & 6l & 0 & 0 \\ 6l & 4l^2 & -6l & 2l^2 & 0 & 0 \\ -12 & -6l & 24 & 0 & -12 & 6l \\ 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 \\ 0 & 0 & -12 & -6l & 12 & -6l \\ 0 & 0 & 6l & 2l^2 & -6l & 4l^2 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{pmatrix} = -\frac{fl}{12}\begin{pmatrix} 6 \\ l \\ 6 \\ -l \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -S(0) \\ -M(0) \\ -Q \\ 0 \\ -kv(3) \\ 0 \end{pmatrix}$$

The unknown displacements $(v_2, \theta_2, v_3, \theta_3)$ are obtained by solving the reduced 4×4 problem (note the k term!):

$$\frac{EI}{l^3}\begin{pmatrix} 24 & 0 & -12 & 6l \\ 0 & 8l^2 & -6l & 2l^2 \\ -12 & -6l & 12 + \frac{kl^3}{EI} & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{pmatrix}\begin{pmatrix} v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} -\frac{fl}{2} - Q \\ \frac{fl^2}{12} \\ 0 \\ 0 \end{pmatrix}.$$

This gives, for instance,

$$v_3 = -\frac{(7fL + 40Q)L^3}{128(kL^3 + 3EI)}, \quad \text{which for } f = k = 0 \text{ reduces to} \quad v_3 = -\frac{5}{48}\frac{QL^3}{EI},$$

in agreement with the exact solution.

The reactions $-S(0)$ and $-M(0)$ at the fixed support can then be obtained from the first two equations of the 6×6 system. We find

$$-S(0) = \frac{fL}{2} + Q + kv_3, \qquad -M(0) = \frac{fL^2}{8} + \frac{QL}{2} + kLv_3,$$

in agreement with overal force and moment balance.

Chapter 6

Time-dependent problems

We consider a representative of both second- and fourth-order problems.

6.1 Second-order problem – transient heat flow

$$\frac{\partial}{\partial x}(Ak\frac{\partial T}{\partial x}) - \alpha(T - T_a) + Q = \rho c A\frac{\partial T}{\partial t},$$

where ρ is the mass density, c the specific heat, A the cross-sectional area of the fin, k the thermal conductivity, α the convective heat flow coefficient, T_a the ambient temperature and Q the heat source.
Weak formulation:

$$\int_0^L \left[-Ak\frac{\partial T}{\partial x}\frac{\partial v}{\partial x} - \alpha T v + \bar{Q}v - \rho c A\frac{\partial T}{\partial t}v\right] \mathrm{d}x + [Ak\frac{\partial T}{\partial x}v]_0^L = 0,$$

where $\bar{Q} = Q + \alpha T_a$.
Interpolation: $u(x,t) = \sum_{j=1}^n N_j(x)T_j(t)$.
Galerkin: $v = N_i \quad (i = 1, ..., n)$.
FE equation (first-order ODE!):

$$M\dot{u} + Ku = f_l + f_b,$$

where

$$u = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix}, \qquad K_{ij} = \int_0^L \frac{\partial N_i}{\partial x}Ak\frac{\partial N_j}{\partial x}\mathrm{d}x + \alpha\int_0^L N_iN_j\mathrm{d}x,$$

$$M_{ij} = \int_0^L \rho c A N_i N_j \mathrm{d}x, \quad f_{l_i} = \int_0^L \bar{Q}N_i\mathrm{d}x, \quad f_{b_i} = [Ak\frac{\partial u}{\partial x}N_i]_0^L = -[AqN_i]_0^L.$$

The new matrix M is called the **mass matrix**.

For a 2-node element, with shape functions $N_1 = 1 - \frac{x}{L}$, $N_2 = \frac{x}{L}$, these element data become (for a uniform bar)

$$K = \frac{Ak}{L}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \frac{\alpha L}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad M = \frac{\rho c A L}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad f_l = \frac{\bar{Q}L}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix},$$
$$f_b = \begin{pmatrix} Aq(0) \\ -Aq(L) \end{pmatrix}.$$

6.2 Fourth-order problem – Euler-Bernoulli beam

$$\frac{\partial^2}{\partial x^2}(EI\frac{\partial^2 v}{\partial x^2}) + \rho A\frac{\partial^2 v}{\partial t^2} = f.$$

Weak formulation (w arbitrary test function):

$$\int_0^L EI\frac{\partial^2 v}{\partial x^2}\frac{\partial^2 w}{\partial x^2}\mathrm{d}x + \int_0^L \rho A\frac{\partial^2 v}{\partial t^2}w\mathrm{d}x = \int_0^L fw\mathrm{d}x + [Sw]_0^L + [M\frac{\partial w}{\partial x}]_0^L.$$

Interpolation: $v(x,t) = \sum_j N_j(x)q_j(t)$, $q = (v_1, \theta_1, v_2, \theta_2, ...)^T$.
Galerkin: $w = N_i$.
FE equation (second-order ODE):

$$M\ddot{q} + Kq = f_l + f_b,$$

where

$$K_{ij} = \int_0^L \frac{d^2 N_i}{dx^2}EI\frac{d^2 N_j}{dx^2}\mathrm{d}x, \qquad M_{ij} = \int_0^L \rho A N_i N_j \mathrm{d}x,$$
$$f_{l_i} = \int_0^L fN_i\mathrm{d}x, \qquad f_{b_i} = [SN_i]_0^L + \left[M\frac{dN_i}{dx}\right]_0^L.$$

For the standard 2-node beam element, with cubic (Hermite) shape functions N_i, the mass matrix is (for a uniform beam, i.e., A and ρ constant)

$$M = \frac{\rho A L}{420}\begin{pmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{pmatrix}.$$

Example (periodically-forced fixed-fixed beam):
Point force at $x = a$: $f(x,t) = P\delta(x-a)\sin\omega t$.
We take the beam to be uniform and to have length L. We consider a mesh of two elements of length $l = L/2$.
Element 1: M^1, K^1 and (assuming $a < L/2$)

$$f_{l_i}^1 = \int_0^l fN_i\mathrm{d}x = P\sin\omega t\int_0^l \delta(x-a)N_i(x)\mathrm{d}x = PN_i(a)\sin\omega t.$$

Element 2: $M^2 = M^1$, $K^2 = K^1$ and $f_l^2 = 0$.
Boundary conditions: $v_1 = 0 = \theta_1$, $v_3 = 0 = \theta_3$
Reduced mass matrix:

$$M = \frac{\rho A l}{420}\begin{pmatrix} 156+156 & -6l+6l \\ -6l+6l & 4l^2+4l^2 \end{pmatrix} = \frac{\rho A l}{105}\begin{pmatrix} 78 & 0 \\ 0 & 2l^2 \end{pmatrix}$$

Reduced stiffness matrix:

$$K = \frac{EI}{l^3}\begin{pmatrix} 12+12 & -6l+6l \\ -6l+6l & 4l^2+4l^2 \end{pmatrix} = \frac{8EI}{l^3}\begin{pmatrix} 3 & 0 \\ 0 & l^2 \end{pmatrix}$$

Reduced load and boundary vectors:

$$f_l = P\sin\omega t\begin{pmatrix} N_3(a) \\ N_4(a) \end{pmatrix}, \qquad f_b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Thus we have

$$\frac{\rho Al}{105}\begin{pmatrix} 78 & 0 \\ 0 & 2l^2 \end{pmatrix}\begin{pmatrix} \ddot{v}_2 \\ \ddot{\theta}_2 \end{pmatrix} + \frac{8EI}{l^3}\begin{pmatrix} 3 & 0 \\ 0 & l^2 \end{pmatrix}\begin{pmatrix} v_2 \\ \theta_2 \end{pmatrix} = P\sin\omega t\begin{pmatrix} N_3(a) \\ N_4(a) \end{pmatrix},$$

(i.e., a system of two periodically-forced uncoupled harmonic oscillators), which can be solved by setting

$$\begin{pmatrix} v_2(t) \\ \theta_2(t) \end{pmatrix} = \begin{pmatrix} \bar{v}_2 \\ \bar{\theta}_2 \end{pmatrix}\sin\omega t.$$

We also consider free vibrations (i.e., no forcing, $P = 0$):

$$\frac{\rho Al}{105}\begin{pmatrix} 78 & 0 \\ 0 & 2l^2 \end{pmatrix}\begin{pmatrix} \ddot{v}_2 \\ \ddot{\theta}_2 \end{pmatrix} + \frac{8EI}{l^3}\begin{pmatrix} 3 & 0 \\ 0 & l^2 \end{pmatrix}\begin{pmatrix} v_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

These are two uncoupled equations, i.e., the physical coordinates (v_2, θ_2) are *normal coordinates* in this case (this is not a general property!). We can write

$$\ddot{v}_2 + \omega_1^2 v_2 = 0, \qquad \ddot{\theta}_2 + \omega_2^2\theta_2 = 0,$$

with *natural frequencies* ω_1 and ω_2 given by

$$\begin{aligned}
\omega_1^2 &= \frac{24EI}{l^3}\frac{105}{78\rho Al} = 32\frac{EI}{\rho Al^4} = 512\frac{EI}{\rho AL^4} \quad \text{(first mode)}, \\
\omega_2^2 &= \frac{4EI}{l^3}\frac{105}{\rho Al} = 420\frac{EI}{\rho Al^4} = 6720\frac{EI}{\rho AL^4} \quad \text{(second mode)}.
\end{aligned}$$

Chapter 7

Nonlinear Problems

7.1 Radiative Heat Transfer

We again consider heat flow in a fin but now include the effect of heat transfer through radiation (which goes as T^4). Assuming that the temperature at the left end ($x = 0$) is kept at $T = T_0$ and that the flux at the right end ($x = L$) is prescribed at $q = q_L$, we have the boundary-value problem

$$\frac{\mathrm{d}}{\mathrm{d}x}(Ak\frac{\mathrm{d}T}{\mathrm{d}x}) - A_s\sigma\left(T^4 - T_\infty^4\right) + Q = 0,$$
$$T(0) = T_0, \quad q(L) = -k\frac{\mathrm{d}T}{\mathrm{d}x}(L) = q_L,$$

where T_∞ is the ambient temperature, A_s is the circumference of the fin and σ is the Stefan-Boltzmann constant.

We solve this problem on a uniform mesh of two 2-node elements of length l (i.e., $2l = L$).

Weak formulation + Galerkin:

$$\int_0^l \left(-\frac{\mathrm{d}v}{\mathrm{d}x}Ak\frac{\mathrm{d}T}{\mathrm{d}x} - A_s\sigma T^4 v + \bar{Q}v\right)\mathrm{d}x - [Aqv]_0^l = 0$$

$$\Leftrightarrow \quad -\sum_j \int_0^l \frac{\mathrm{d}N_i}{\mathrm{d}x}Ak\frac{\mathrm{d}N_j}{\mathrm{d}x}\mathrm{d}xT_j - A_s\sigma\int_0^l N_i\left(\sum_j N_jT_j\right)^4 \mathrm{d}x + \int_0^l \bar{Q}N_i\mathrm{d}x - [AqN_i]_0^l = 0$$

where $\bar{Q} = Q + A_s\sigma T_\infty^4$. The nonlinear term gives rise to the following term in the FE equation:

$$-A_s\sigma\begin{pmatrix} \int_0^l N_1(N_1T_1 + N_2T_2)^4\,\mathrm{d}x \\ \int_0^l N_2(N_1T_1 + N_2T_2)^4\,\mathrm{d}x \end{pmatrix}$$

$$= -A_s\sigma\begin{pmatrix} \int N_1^5 & 4\int N_1^4N_2 & 6\int N_1^3N_2^2 & 4\int N_1^2N_2^3 & \int N_1N_2^4 \\ \int N_1^4N_2 & 4\int N_1^3N_2^2 & 6\int N_1^2N_2^3 & 4\int N_1^1N_2^4 & \int N_2^5 \end{pmatrix}\begin{pmatrix} T_1^4 \\ T_1^3T_2 \\ T_1^2T_2^2 \\ T_1T_2^3 \\ T_2^4 \end{pmatrix},$$

which, with the shape functions $N_1 = 1 - x/l$, $N_2 = x/l$, becomes

$$-lA_s\sigma \begin{pmatrix} \frac{1}{6} & \frac{2}{15} & \frac{1}{10} & \frac{1}{15} & \frac{1}{30} \\ \frac{1}{30} & \frac{1}{15} & \frac{1}{10} & \frac{2}{15} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} T_1^4 \\ T_1^3T_2 \\ T_1^2T_2^2 \\ T_1T_2^3 \\ T_2^4 \end{pmatrix},$$

where we have used the integration formula

$$\int_0^L N_1^n N_2^m \mathrm{d}x = \frac{n!\,m!}{(n+m+1)!}L$$

(the shape functions N_i are 'length coordinates' analogous to the 'area coordinates' for triangular elements in 2D).

The element stiffness matrix and load vector are the usual

$$K = \frac{Ak}{l}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad f_l = \frac{\bar{Q}l}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Assembly gives for the 2-element mesh:

$$lA_s\sigma \begin{pmatrix} \frac{1}{6} & \frac{2}{15} & \frac{1}{10} & \frac{1}{15} & \frac{1}{30} & 0 & 0 & 0 & 0 \\ \frac{1}{30} & \frac{1}{15} & \frac{1}{10} & \frac{2}{15} & \frac{1}{3} & \frac{2}{15} & \frac{1}{10} & \frac{1}{15} & \frac{1}{30} \\ 0 & 0 & 0 & 0 & \frac{1}{30} & \frac{1}{15} & \frac{1}{10} & \frac{2}{15} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} T_1^4 \\ T_1^3T_2 \\ T_1^2T_2^2 \\ T_1T_2^3 \\ T_2^4 \\ T_2^3T_3 \\ T_2^2T_3^2 \\ T_2T_3^3 \\ T_3^4 \end{pmatrix}$$

$$+\frac{Ak}{l}\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \frac{\bar{Q}l}{2}\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} Aq(0) \\ 0 \\ -Aq(L) \end{pmatrix},$$

where we note that T_2^4 is the only monomial common to both elements.

After applying the boundary conditions, $T_1 = T_0$, $q(L) = q_L$, this represents three nonlinear algebraic equations for T_2, T_3 and $q(0)$. The second and third can be solved for T_2 and T_3 first, after which $q(0)$ can be obtained from the first equation.

7.2 The Newton-Raphson Method

A good method for solving nonlinear algebraic equations is the Newton-Raphson method. Recall that for a single equation, $f(x) = 0$, this method gives the iterative scheme

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

which can be started from an initial guess x_1.

Convergence $x_k \to \bar{x}$ of the Newton-Raphson method (to a nonsingular root $\bar{x}$, i.e., an $\bar{x}$ such that $f(\bar{x}) = 0$ and $f'(\bar{x}) \neq 0$) is (at least) quadratic (as follows by Taylor expansion), which means that the number of correct digits, ultimately, doubles in each iteration. Nonlinear equations, however, may have multiple solutions and the initial guess x_1 has to be good enough for the scheme to converge to the desired solution.

For two equations, $F(x,y) = (f_1(x,y), f_2(x,y)) = 0$, of two unknowns, (x,y), the above scheme generalises to

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - DF(x_k,y_k)^{-1} \begin{pmatrix} f_1(x_k,y_k) \\ f_2(x_k,y_k) \end{pmatrix}.$$

Here, DF is the derivative of F, i.e., the matrix of partial derivatives:

$$DF(x,y) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix}.$$

Exercise: In special cases Newton-Raphson may converge faster than quadratic. For instance, if $f''(\bar{x}) = 0$ then convergence to $\bar{x}$ is cubic. Verify this by computing a few iterates for the root $\bar{x} = 0$ of $f(x) = x + x^3$. Back up your observations with a Taylor expansion. Give an example with quartic convergence.

Chapter 8

Case Study with Nonlinearity: Solute Transport in Soil

Governing equation (advection-dispersion-reaction equation):

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial x} + D_L\frac{\partial^2 C}{\partial x^2} + pC(1-C).$$

C is the solute concentration. Its rate of change $\frac{\partial C}{\partial t}$ is governed by three physical processes.
$-v\frac{\partial C}{\partial x}$ is the advection term due to ground water flow with velocity v,
$D_L\frac{\partial^2 C}{\partial x^2}$ is the dispersion term (for transport due to a concentration gradient $\frac{\partial C}{\partial x}$, analogous to conduction in heat flow),
$pC(1-C)$ is the reaction term (this is a **nonlinear** term because it takes two solute particles to chemically react to another species).
Weak formulation:

$$\int_0^L (-v\frac{\partial C}{\partial x}w + D_L\frac{\partial C}{\partial x}\frac{\partial w}{\partial x} + pCw - pC^2w - \frac{\partial C}{\partial t}w)\mathrm{d}x - [D_L\frac{\partial C}{\partial x}w]_0^L = 0$$

2-node element: $C = N_1C_1 + N_2C_2$, with

$$N_1 = 1 - \frac{x}{L}, \qquad N_2 = \frac{x}{L}.$$

Galerkin: $w = N_1, N_2$.
FE equation:

$$[-v\sum_{j=1}^{2}\int_0^L \frac{\partial N_j}{\partial x}N_i\mathrm{d}x + D_L\sum_{j=1}^{2}\int_0^L \frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}\mathrm{d}x + p\sum_{j=1}^{2}\int_0^L N_iN_j\mathrm{d}x]C_j$$
$$-\int_0^L p(N_1C_1+N_2C_2)^2N_i\mathrm{d}x - \sum_{j=1}^{2}\int_0^L N_iN_j\mathrm{d}x\dot{C}_j = [JN_i]_0^L,$$

where we have used Fick's Law $J = -D_L\frac{\partial C}{\partial x}$ (analogous to Fourier's Law in heat flow).

We know (or compute)

$$\int_0^L \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}\mathrm{d}x \to \tfrac{1}{L}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}$$
$$\int_0^L N_i N_j \mathrm{d}x \to \tfrac{L}{6}\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}$$
$$\int_0^L N_i \frac{\partial N_j}{\partial x}\mathrm{d}x \to \tfrac{1}{2}\begin{pmatrix}-1 & 1\\ -1 & 1\end{pmatrix}$$

(Note that this last matrix is nonsymmetric. Indeed, the advection term in the equation, having a single derivative, is non-self-adjoint. As a consequence, the advection-dispersion-reaction equation has no variational principle. However, we can still do a finite-element analysis using the Galerkin weak formulation.)

Nonlinear term

$$\begin{aligned}
&\int_0^L (N_1C_1 + N_2C_2)^2 N_i \mathrm{d}x\\
\to &\int_0^L (N_1^2C_1^2 + 2N_1N_2C_1C_2 + N_2^2C_2^2)\begin{pmatrix}N_1\\ N_2\end{pmatrix}\mathrm{d}x\\
= &\int_0^L \begin{pmatrix}N_1^3 & 2N_1^2N_2 & N_1N_2^2\\ N_1^2N_2 & 2N_1N_2^2 & N_2^3\end{pmatrix}\begin{pmatrix}C_1^2\\ C_1C_2\\ C_2^2\end{pmatrix}\mathrm{d}x\\
= &L\begin{pmatrix}\frac{1}{4} & \frac{1}{6} & \frac{1}{12}\\ \frac{1}{12} & \frac{1}{6} & \frac{1}{4}\end{pmatrix}\begin{pmatrix}C_1^2\\ C_1C_2\\ C_2^2\end{pmatrix}.
\end{aligned}$$

Putting everything together:

$$\begin{aligned}
&-\frac{v}{2}\begin{pmatrix}-1 & 1\\ -1 & 1\end{pmatrix}\begin{pmatrix}C_1\\ C_2\end{pmatrix} - \frac{D_L}{L}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}\begin{pmatrix}C_1\\ C_2\end{pmatrix} + \frac{pL}{6}\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}\begin{pmatrix}C_1\\ C_2\end{pmatrix}\\
&-\frac{pL}{12}\begin{pmatrix}3 & 2 & 1\\ 1 & 2 & 3\end{pmatrix}\begin{pmatrix}C_1^2\\ C_1C_2\\ C_2^2\end{pmatrix} - \frac{L}{6}\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}\begin{pmatrix}\dot{C}_1\\ \dot{C}_2\end{pmatrix} = \begin{pmatrix}-J(0)\\ J(L)\end{pmatrix}.
\end{aligned}$$

This can be written in standard form by inverting the mass matrix, $\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}^{-1} = \frac{1}{3}\begin{pmatrix}2 & -1\\ -1 & 2\end{pmatrix}$:

$$\begin{aligned}
\begin{pmatrix}\dot{C}_1\\ \dot{C}_2\end{pmatrix} &= [-\frac{v}{L}\begin{pmatrix}-1 & 1\\ -1 & 1\end{pmatrix} - \frac{D_L}{L^2}\begin{pmatrix}3 & -3\\ -3 & 3\end{pmatrix} + p\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}]\begin{pmatrix}C_1\\ C_2\end{pmatrix}\\
&-\frac{p}{6}\begin{pmatrix}5C_1^2 + C_1C_2 - C_2^2\\ -C_1^2 + C_1C_2 + 5C_2^2\end{pmatrix} - \frac{2}{L}\begin{pmatrix}-2J(0) - J(L)\\ J(0) + 2J(L)\end{pmatrix}.
\end{aligned} \tag{8.1}$$

This is a nonlinear ODE with unknowns $C_1, C_2, J(0), J(L)$.
Possible initial/boundary conditions: C and/or flux J given at $x = 0$ and $x = L$. If C is given (at all t), then C_1 and C_2 are known as funtions of t, so we can obtain $J(0)$ and $J(L)$ directly from the equation.

If $J(0)$ and $J(L)$ are given as functions of t, then we need initial conditions $C_1(0),\ C_2(0)$ to integrate the equation.

The nonlinear ODE (8.1) can be integrated by any (finite-difference) time-stepping method (preferably an unconditionally stable second-order accurate method). For stationary (i.e., time-independent) solutions we can set the time-derivatives $\dot{C}_1$ and $\dot{C}_2$ in (8.1) equal to zero. This leads to a coupled system of nonlinear algebraic equations in C_1 and C_2 that can be solved by the Newton-Raphson method.