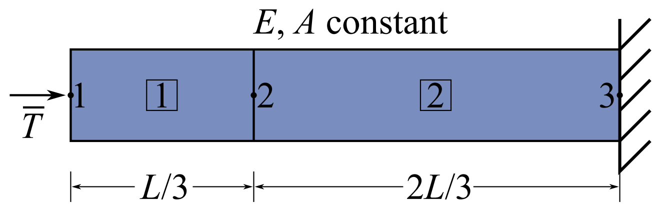
Example: One-Dimensional Finite Element



the bar shown above is one-dimensional: all forces and displacements are axial to the bar

it is fixed at the right end and loaded with a traction $\bar{T}$ at the free end

it is divided into two one-dimensional elements, associated with three nodes

use the finite element method to find the nodal displacements for this bar

Example: One-Dimensional Finite Element

evaluate the integral that provides the element stiffness matrix:

$$\mathbf{K}_{\text{elem}} = \int_{x_1}^{x_2} \mathbf{H}^T AE\mathbf{H}\,\mathrm{d}x$$

in this case, E and A are constant and come outside the integral; expressing $\mathbf{H}^T$ and $\mathbf{H}$ in full and using $x_1 = 0$ and $x_2 = \ell$ gives:

$$\begin{aligned}\mathbf{K}_{\text{elem}} &= AE\int_0^{\ell}\begin{bmatrix} -\frac{1}{\ell} \\ \frac{1}{\ell}\end{bmatrix}\begin{bmatrix} -\frac{1}{\ell} & \frac{1}{\ell}\end{bmatrix}\mathrm{d}x \\ &= AE\int_0^{\ell}\begin{bmatrix} \frac{1}{\ell^2} & -\frac{1}{\ell^2} \\ -\frac{1}{\ell^2} & \frac{1}{\ell^2}\end{bmatrix}\mathrm{d}x \\ &= \frac{AE}{\ell}\begin{bmatrix} 1 & -1 \\ -1 & 1\end{bmatrix}\end{aligned}$$

Example: One-Dimensional Finite Element

next, calculate the element stiffness matrices for each of the two elements:

$$\mathbf{K}_1 = \frac{3AE}{L}\cdot \begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & -1 \\ 2 & -1 & 1 \end{array}$$

$$\mathbf{K}_2 = \frac{3AE}{2L}\cdot \begin{array}{c|cc} & 2 & 3 \\ \hline 2 & 1 & -1 \\ 3 & -1 & 1 \end{array}$$

combine the two element matrices into a single structural stiffness matrix:

$$\mathbf{K} = \frac{3AE}{L} \cdot \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1 & -1 & 0 \\ 2 & -1 & \frac{3}{2} & -\frac{1}{2} \\ 3 & 0 & -\frac{1}{2} & \frac{1}{2} \end{array}$$

Example: One-Dimensional Finite Element

determine the force vector and the displacement vector; first the force vector:

$$\sum^{\text{elems}} \left(\int_{x_1}^{x_2} \cancelto{0}{\mathbf{N}^T B}\, \mathrm{d}x - \left(\mathbf{N}^T A \bar{T} \right)_{x=0} \right)$$

the transpose of the element shape function, $\mathbf{N}^T$, is:

$$\mathbf{N}^T = \begin{bmatrix} \frac{\ell - x}{\ell} \\ \frac{x}{\ell} \end{bmatrix}$$

so for element 1

$$\mathbf{f}_1 = \left(\mathbf{N}^T A \bar{T} \right)_{x=0} = \begin{bmatrix} A\bar{T} \\ 0 \end{bmatrix}$$

and for the full force vector:

$$\mathbf{f} = \begin{bmatrix} A\bar{T} \\ 0 \\ f_3 \end{bmatrix}$$

Example: One-Dimensional Finite Element

the displacement vector is:

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ 0 \end{bmatrix}$$

note that d_1, d_2 and f_3 are unknown and must be solved; for each pair d_i, f_i, one of the pair must be unknown

the problem to solve is:

$$\mathbf{f} = \mathbf{K}\mathbf{d}$$

Example: One-Dimensional Finite Element

part of the displacement vector, **d**, is zero, so partition the problem to solve the smallest linear system possible:

$$\begin{bmatrix} A\bar{T} \\ 0 \end{bmatrix} = \frac{3AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & \frac{3}{2} \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

solving:

$$d_1 = \frac{3}{2}d_2; \quad A\bar{T} = \frac{3AE}{L}\left(\frac{3}{2}d_2 - d_2\right) \Longrightarrow d_2 = \frac{2\bar{T}L}{3E}; \quad d_1 = \frac{\bar{T}L}{E}$$

Example: One-Dimensional Finite Element

if desired, solve for the reaction force, f_3, using the full displacement vector and the full stiffness matrix:

$$\begin{bmatrix} A\bar{T} \\ 0 \\ f_3 \end{bmatrix} = \frac{3AE}{L} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} \frac{\bar{T}L}{E} \\ \frac{2\bar{T}L}{3E} \\ 0 \end{bmatrix}$$

$$f_3 = \frac{3AE}{L} \cdot 0 \cdot \frac{\bar{T}L}{E} + \frac{3AE}{L} \cdot -\frac{1}{2} \cdot \frac{2\bar{T}L}{3E} + \frac{3AE}{L} \cdot \frac{1}{2} \cdot 0 = -A\bar{T}$$

3.8 Two-Dimensional Solid Elements

The Process of Finite Element Analysis

there is a standard set of steps to set up a finite element analysis

1. develop the strong form of the equilibrium equation
2. convert the strong form to the weak form
3. discretize the domain into finite elements
4. approximate the discretized weak form using the shape functions and their derivatives

Strong Form of Linear Elasticity in Two Dimensions

to develop a two-dimensional finite element use the strong form for a vector-valued function (displacement) in two dimensions

the strong form consists of:

1. kinematic conditions that relate strain and displacement
2. equilibrium conditions
3. a constitutive relation: linear elasticity
4. stress and traction relations to ensure that the stress state is consistent with the boundary conditions

this is identical in process to the one-dimensional elements: combining these four considerations creates the strong form

Strong Form of Linear Elasticity in Two Dimensions

for a linear analysis, assume that there are neither material nor geometric non-linearities; the displacements and rotations therefore must be small

the kinematic relation between strain and displacement is the linearised form of the strains:

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$

using $\gamma_{12} = 2\epsilon_{12}$:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & \gamma_{12} \end{bmatrix}^T = \boldsymbol{\nabla}_S \mathbf{u}$$

where the symmetric gradient matrix operator is:

$$\boldsymbol{\nabla}_S = \begin{bmatrix} \dfrac{\partial}{\partial x_1} & 0 \\ 0 & \dfrac{\partial}{\partial x_2} \\ \dfrac{\partial}{\partial x_2} & \dfrac{\partial}{\partial x_1} \end{bmatrix}$$

and:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Strong Form of Linear Elasticity in Two Dimensions

the equilibrium relation is Euler's equation for equilibrium:

$$\sigma_{ji,j} + b_i = 0$$

writing this out in full for two dimensions gives:

$$\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{21}}{\partial x_2} + b_1 = 0$$

$$\frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + b_2 = 0$$

taking advantage of the symmetry of σ_{ij} and the fact that $\sigma_{12} = \sigma_{21}$, write this in matrix form as:

$$\boldsymbol{\nabla}_S^T \boldsymbol{\sigma} + \overline{\mathbf{B}} = \boldsymbol{\nabla}_S^T \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} + \overline{\mathbf{B}} = 0$$

where $\boldsymbol{\nabla}_S^T$ is the transpose of the symmetric gradient matrix operator

Strong Form of Linear Elasticity in Two Dimensions

the constitutive equation is Hooke's Law:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$$

using the vector representation of stress, for plane stress:

$$\mathbf{D} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

and for plane strain:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

Strong Form of Linear Elasticity in Two Dimensions

the traction on a surface is given by Cauchy's law:

$$t_i = \sigma_{ji}\nu_j$$

it is standard in the finite element literature, like the composites literature, to use a vector representation of stress:

$$\boldsymbol{\sigma}^T = [\sigma_{11} \quad \sigma_{22} \quad \sigma_{12}] \equiv \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$$

using this, in vector notation the relation of the surface traction to the local stress state is:

$$\overline{\mathbf{T}} = \boldsymbol{\sigma}\boldsymbol{\nu}$$

Strong Form of Linear Elasticity in Two Dimensions

the summary of the strong form in two dimensions is:

1. *strain-displacement:*

$$\boldsymbol{\epsilon} = \boldsymbol{\nabla}_S \mathbf{u}$$

2. *equilibrium:*

$$\boldsymbol{\nabla}_S^T \boldsymbol{\sigma} + \overline{\mathbf{B}} = 0$$

3. *constitutive law:*

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$$

4. *boundary conditions:*

$$\overline{\mathbf{T}} = \boldsymbol{\sigma}\boldsymbol{\nu}$$

combining 1, 2 and 3 produces:

$$\boldsymbol{\nabla}_S^T (\mathbf{D}\boldsymbol{\nabla}_S \mathbf{u}) + \overline{\mathbf{B}} = 0$$

Weak Form of Linear Elasticity in Two Dimensions

convert the strong form to the weak form by multiplying the strong form by a weight function (a variation) and integrating by parts

the weight function must be vector-valued to be consistent with the displacement solution function **u**:

$$\delta w_i = \delta \mathbf{w} = \delta \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \delta w_1 \\ \delta w_2 \end{bmatrix}$$

the weight function is zero at all surfaces with imposed displacement boundary conditions

this must be done for each dimension:

$$\int_\Omega \delta w_i \sigma_{ji,j} \, \mathrm{d}\Omega + \int_\Omega \delta w_i b_i \, \mathrm{d}\Omega = 0 \qquad \text{no summation on } i$$

$$\int_\Gamma \delta w_i \left(t_i - \sigma_{ji} \nu_j \right) \, \mathrm{d}\Gamma = 0 \qquad \text{no summation on } i$$

Weak Form of Linear Elasticity in Two Dimensions

apply Gauss's divergence theorem, integrate by parts, evaluate boundary conditions and add the equations for each dimension together into a single equation:

$$\int_\Omega \delta w_{i,j} \sigma_{ij} \, \mathrm{d}\Omega = \int_\Gamma \delta w_i t_i \, \mathrm{d}\Gamma + \int_\Omega \delta w_i b_i \, \mathrm{d}\Omega$$

expand the integrand of the first integral as:

$$\begin{aligned} \delta w_{i,j} \sigma_{ij} &= \frac{\partial \delta w_1}{\partial x_1} \sigma_{11} + \frac{\partial \delta w_1}{\partial x_2} \sigma_{12} + \frac{\partial \delta w_2}{\partial x_1} \sigma_{12} + \frac{\partial \delta w_2}{\partial x_2} \sigma_{22} \\ &= \begin{bmatrix} \frac{\partial \delta w_1}{\partial x_1} & \frac{\partial \delta w_2}{\partial x_2} & \left(\frac{\partial \delta w_1}{\partial x_2} + \frac{\partial \delta w_2}{\partial x_1} \right) \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} \\ &= \left(\boldsymbol{\nabla}_S \delta \mathbf{w} \right)^T \boldsymbol{\sigma} \end{aligned}$$

using this:

$$\int_\Omega \left(\boldsymbol{\nabla}_S \delta \mathbf{w} \right)^T \boldsymbol{\sigma} \, \mathrm{d}\Omega = \int_\Gamma \delta \mathbf{w}^T \overline{\mathbf{T}} \, \mathrm{d}\Gamma + \int_\Omega \delta \mathbf{w}^T \overline{\mathbf{B}} \, \mathrm{d}\Omega$$

Weak Form of Linear Elasticity in Two Dimensions

finally, employ Hooke's Law and the strain-displacement relation:

$$\int_\Omega \left(\boldsymbol{\nabla}_S \delta \mathbf{w} \right)^T \mathbf{D} \boldsymbol{\nabla}_S \mathbf{u} \, \mathrm{d}\Omega = \int_\Gamma \delta \mathbf{w}^T \overline{\mathbf{T}} \, \mathrm{d}\Gamma + \int_\Omega \delta \mathbf{w}^T \overline{\mathbf{B}} \, \mathrm{d}\Omega$$

this is the weak form of the two-dimensional linear elastic system for displacements

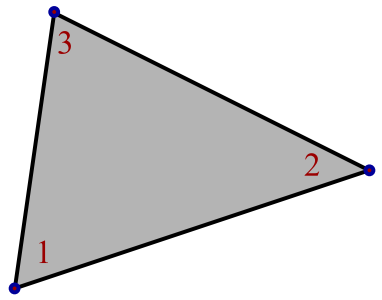
these make the implicit assumption that the system is in plane stress or plane strain and is one unit thick

this will be used to perform the finite element discretisation

3.9 Two-Dimensional Solid Triangular Elements

Two-Dimensional Triangular Element

the simplest two-dimensional element is the three-node triangular element



a disadvantage is that the three-node triangular element is not very accurate: it tends to be much too stiff

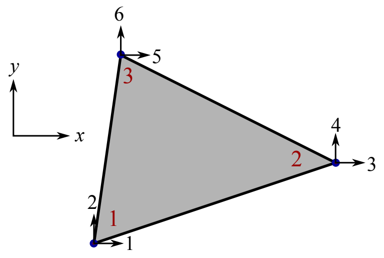
note that the order of the nodes is *always* anticlockwise; because the nodes are numbered 1 to 3, use x, y coordinates and u, v displacements to keep the notation clear

Degrees of Freedom

before beginning the derivation of the triangular element, an explanation of *degrees of freedom* is necessary

a two-dimensional element will have two degrees of freedom at each node: this means that at each node there are two displacements, one in the x-direction and one in the y-direction, that are called u and v

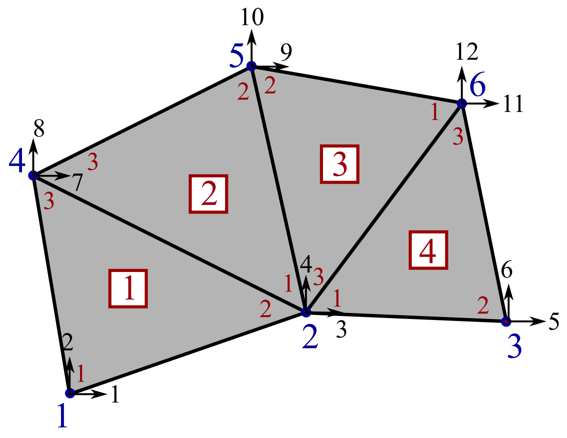
at each node there are also two forces, again one in the x-direction and one in the y-direction



each degree of freedom has a local number from 1 to 6 for use when referring to the element degrees of freedom

Degrees of Freedom

to model a domain, several elements are combined into a mesh



each node must have a unique global node number, and each degree of freedom must have a unique global degree of freedom number

Two-Dimensional Triangular Element

to begin developing the triangular element, consider a scalar function T (imagine temperature) that is to be approximated within the triangular element; the approximation uses three constants, to be consistent with the number of nodes in the element:

$$T(x,y) = \alpha_0 + \alpha_1 x + \alpha_2 y = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \mathbf{p}\boldsymbol{\alpha}$$

here, $T(x,y)$ is the approximated value of T at some location (x,y) within the element

this approximation will be used to construct shape functions for a linear triangular element; use the term *linear* to refer to the order of the approximation functions (and hence the shape functions) that we use

Two-Dimensional Triangular Element

express the nodal values of the function T in terms of the constants α_i and the nodal locations:

$$T_1 = \alpha_0 + \alpha_1 x_1 + \alpha_2 y_1$$

$$T_2 = \alpha_0 + \alpha_1 x_2 + \alpha_2 y_2$$

$$T_3 = \alpha_0 + \alpha_1 x_3 + \alpha_2 y_3$$

here, x_1 is the x coordinate of node 1, x_2 is the x coordinate of node 2, and so on rewrite this in vector - matrix form:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \mathbf{T} = \mathbf{M}\alpha$$

inverting this gives:

$$\alpha = \mathbf{M}^{-1}\mathbf{T}$$

Two-Dimensional Triangular Element

this is an expression for $\boldsymbol{\alpha}$ that can be substituted into the equation for $T(x, y)$:

$$T\,(x, y) = \mathbf{pM}^{-1}\mathbf{T}$$

similar to the one-dimensional element, express the function T as a shape function multiplied by the nodal values of the function:

$$T\,(x, y) = \mathbf{NT}$$

where:

$$\mathbf{N} = \mathbf{pM}^{-1}$$

this requires an expression for $\mathbf{M}^{-1}$:

$$\mathbf{M}^{-1} = \frac{1}{2A}\begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix}$$

where:

$$2A = \det(\mathbf{M}) = (x_2y_3 - x_3y_2) - (x_1y_3 - x_3y_1) + (x_1y_2 - x_2y_1)$$

Triangular Element Shape Functions

evaluating the last expressions for $\mathbf{N}$ produces the three shape functions:

$$N_1 = \frac{1}{2A}\Big((x_2y_3 - x_3y_2) + (y_2 - y_3)\,x + (x_3 - x_2)\,y \Big)$$

$$N_2 = \frac{1}{2A}\Big((x_3y_1 - x_1y_3) + (y_3 - y_1)\,x + (x_1 - x_3)\,y \Big)$$

$$N_3 = \frac{1}{2A}\Big((x_1y_2 - x_2y_1) + (y_1 - y_2)\,x + (x_2 - x_1)\,y \Big)$$

the derivative of the shape functions is:

$$\mathbf{H} = \frac{1}{2A}\begin{bmatrix} (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix}$$

note that $\mathbf{H}$ is not a function of x or y; this makes it very easy to integrate during the creation of the element stiffness matrix

Triangular Element - Vector Functions

note that this is only for a scalar function in two dimensions; displacement has two components in two dimensions, so it is a vector-valued function

for vector-valued functions, such as $\mathbf{u} = [u, v]^T$, both components must be interpolated and discretized

each component of **u** can be treated as an independent scalar variable: interpolate u and v separately

in principle, it is possible to use different shape functions for each component of **u**; however, this is almost never done and the same shape functions are used for all of the components of the vector

for both components of **u**, use linear interpolations and hence linear shape functions

Triangular Element - Vector Functions

each node has two components of displacement and hence two degrees of freedom, so the vector of nodal displacements has six components:

$$\mathbf{d} = \begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_3 & v_3 \end{bmatrix}^T$$

the element shape function matrix is:

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}$$

such that:

$$\mathbf{u}(x, y) \approx \mathbf{Nd}$$

recall the weak form of the governing equations:

$$\int_\Omega (\boldsymbol{\nabla}_S \delta\mathbf{w})^T \, \mathbf{D}\boldsymbol{\nabla}_S \mathbf{u} \, \mathrm{d}\Omega = \int_\Gamma \delta\mathbf{w}^T \overline{\mathbf{T}} \, \mathrm{d}\Gamma + \int_\Omega \delta\mathbf{w}^T \overline{\mathbf{B}} \, \mathrm{d}\Omega$$

and replace the integrands with their discretised approximations

Triangular Element - Vector Functions

first, recall the appropriate treatment for the strain:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \gamma_{12} \end{bmatrix} = \boldsymbol{\nabla}_S \mathbf{u} \approx \boldsymbol{\nabla}_S \mathbf{Nd} = \mathbf{Hd}$$

again, this requires a representation for the derivatives of **N**:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix}$$

and **H** is the strain - displacement relation

Triangular Element - Vector Functions

additionally, this requires a representation for the weight function $\delta\mathbf{w}$;

employ a Galerkin method, and approximate it with $\delta\mathbf{w}(x, y) \approx \mathbf{N}\mathbf{w}$ where **w** is the value of the weight function at the nodes:

$$(\boldsymbol{\nabla}_S\,\mathbf{w})^T \approx (\boldsymbol{\nabla}_S\,\mathbf{N}\mathbf{w}) = (\mathbf{H}\mathbf{w})^T = \mathbf{w}^T\mathbf{H}^T$$

substitute the approximations into the weak form and discretize, element by element:

$$\mathbf{w}^T \sum^{\text{elems}} \left(\int_\Omega \mathbf{H}^T\mathbf{D}\mathbf{H}\,\mathrm{d}\Omega\mathbf{d} - \int_\Gamma \mathbf{N}^T\overline{\mathbf{T}}\,\mathrm{d}\Gamma - \int_\Omega \mathbf{N}^T\overline{\mathbf{B}}\,\mathrm{d}\Omega \right) = 0$$

the element stiffness matrix is:

$$\mathbf{K}_e = \int_\Omega \mathbf{H}^T\mathbf{D}\mathbf{H}\,\mathrm{d}\Omega$$

3.10 Gauss Quadrature

Gauss Quadrature

for the two-dimensional triangular element, the integrals that had to be evaluated were all integrals of constants, and hence were unproblematic

in general, the integrals that must be evaluated cannot be integrated in closed form

a fast, accurate numerical method to evaluate the necessary integrals is essential

Gauss quadrature is one of the most efficient methods available to integrate polynomial or nearly polynomial functions

Gauss quadrature will be the standard method in this course for evaluating the required integrals

Gauss Quadrature

Gauss quadrature is used to evaluate integrals:

$$I = \int_a^b f(x)\,\mathrm{d}x$$

where the function $f(x)$ is known and can be evaluated at any arbitrary point, but there is not have a conveniently available closed-form indefinite integral to evaluate

the Gauss quadrature formulae enable numerical integration of such integrals; the process involves evaluation of the function $f(x)$ at certain points, multiplying the function evaluations by weights, and then summing the resulting products

first, map the interval, a to b, over which the integral is to be evaluated onto a parent domain, which is always $\xi \in [-1, 1]$

Gauss Quadrature in One Dimension

the relation between the physical coordinate x and the parent coordinate ξ is:

$$x = \frac{1}{2}(a + b) + \frac{1}{2}\xi\,(b - a)$$

which can be transformed into:

$$x = a\frac{1-\xi}{2} + b\frac{1+\xi}{2} = aN_1(\xi) + bN_2(\xi)$$

which strongly resembles linear shape functions: Gauss quadrature and linear approximations are very closely related

Gauss Quadrature in One Dimension

from evaluating the derivatives:

$$\mathrm{d}x = \frac{1}{2}(b - a)\ \mathrm{d}\xi = \frac{\ell}{2}\,\mathrm{d}\xi = J\,\mathrm{d}\xi$$

where J is the Jacobian determinant given by $(b-a)/2$, which is also half the element length

write the integral as:

$$I = J\int_{-1}^{1} f(\xi)\,\mathrm{d}\xi = J\hat{I}$$

it is this integral $\hat{I}$ that will be approximated by the Gauss quadrature procedure

Gauss Quadrature in One Dimension

approximate the function $\hat{I}$ using a sum of the products of evaluations of the function f and weights:

$$\hat{I} \approx W_1 f(\xi_1) + W_2 f(\xi_2) + \cdots + W_n f(\xi_n) = \begin{bmatrix} W_1 & W_2 & \dots & W_n \end{bmatrix} \begin{bmatrix} f(\xi_1) \\ f(\xi_2) \\ \vdots \\ f(\xi_n) \end{bmatrix} \equiv \mathbf{W}^T\mathbf{f}$$

where W_i are the weight functions; the ξ_i are locations at which the function f is to be evaluated, not different coordinate axes

choose the weights W_i and the locations at which the function is evaluated ξ_i to integrate the highest possible polynomial exactly

the question is, how are the weights and the locations (known as integration points or Gauss points) determined?

Gauss Quadrature in One Dimension

first, approximate the function $f(\xi)$ by the $(m-1)^{\text{th}}$ order polynomial:

$$f(\xi) \approx \alpha_1 + \alpha_2\xi + \alpha_3\xi^2 + \cdots = \begin{bmatrix} 1 & \xi & \xi^2 & \cdots & \xi^{m-1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_m \end{bmatrix} \equiv \mathbf{p}(\xi)\boldsymbol{\alpha}$$

next, express the values of the coefficients α_i in terms of the values of the function $f(\xi)$ at the integration points:

$$\mathbf{f} \equiv \begin{bmatrix} f(\xi_1) \\ f(\xi_2) \\ \vdots \\ f(\xi_n) \end{bmatrix} = \begin{bmatrix} 1 & \xi_1 & \xi_1^2 & \cdots & \xi_1^{m-1} \\ 1 & \xi_2 & \xi_2^2 & \cdots & \xi_2^{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_n & \xi_n^2 & \cdots & \xi_n^{m-1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_m \end{bmatrix} \equiv \mathbf{M}\boldsymbol{\alpha}$$

by combining these:

$$\hat{I} \approx \mathbf{W}^T\mathbf{M}\boldsymbol{\alpha}$$

Gauss Quadrature in One Dimension

this analysis provides the weights and integration points necessary to evaluate polynomials of a certain order exactly; the order of the polynomial that can be evaluated exactly is determined by the number of integration points

if n_{gp} is the number of Gauss integration points, the the order p of the polynomial that can be integrated exactly is:

$$p \leq 2n_{gp} - 1$$

to find the weights and the Gauss points, integrate the polynomial $f(\xi)$:

$$\hat{I} = \int_{-1}^{1} f(\xi)\mathrm{d}\xi \approx \int_{-1}^{1} \left[\begin{array}{ccccc} 1 & \xi & \xi^2 & \cdots & \xi^{m-1} \end{array} \right] \left[\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{array} \right] \mathrm{d}\xi = \left[\begin{array}{cccc} \xi & \frac{\xi^2}{2} & \frac{\xi^3}{3} & \cdots \end{array} \right]_{-1}^{1} \boldsymbol{\alpha}$$

or

$$\hat{I} \approx \left[\begin{array}{ccccc} 2 & 0 & \frac{2}{3} & 0 & \cdots \end{array} \right] \boldsymbol{\alpha} \equiv \hat{\mathbf{P}}\boldsymbol{\alpha}$$

Gauss Quadrature in One Dimension

the integral has been expressed in two ways:

$$\hat{I} \approx \mathbf{W}^T\mathbf{M}\boldsymbol{\alpha}$$

and

$$\hat{I} \approx \hat{\mathbf{P}}\boldsymbol{\alpha}$$

as well, $\hat{\mathbf{P}}$ has been determined through analytical integration

therefore:

$$\mathbf{W}^T\mathbf{M} = \hat{\mathbf{P}} \equiv \left[\begin{array}{ccccc} 2 & 0 & \frac{2}{3} & 0 & \cdots \end{array} \right] \Rightarrow \mathbf{M}^T\mathbf{W} = \hat{\mathbf{P}}^T$$

which is a system of non-linear equations for the Gauss points (**M**) and the weights (**W**)

Gauss Quadrature in One Dimension

solve this system for as many Gauss points as desired; the results are tabulated:

n_{gp}	Gauss point ξ_i	Weight W_i
1	0.0	2.0
2	± 0.5773502693	1.0
3	± 0.7745966692	0.5555555556
	0.0	0.8888888889
4	± 0.8611363116	0.3478548451
	± 0.3399810436	0.6521451549

Gauss Quadrature in Two Dimensions

this enables evaluation of integrals of functions that are polynomial or nearly polynomial, which includes all of the shape functions in this course

in two dimensions, there will be integrals of the form:

$$I = \int_\Omega f(x, y)\, \mathrm{d}\Omega$$

this requires conversion of the physical area, $\mathrm{d}\Omega$, to an area in the parent coordinates (and the reverse operation):

$$\mathrm{d}\Omega = \det \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \mathrm{d}\xi\, \mathrm{d}\eta = |\mathbf{J}|\ \mathrm{d}\xi\, \mathrm{d}\eta$$

where $|\mathbf{J}|$ is the determinant of the Jacobian matrix, often called the Jacobian determinant

Gauss Quadrature in Two Dimensions

re-express the integral as:

$$I = \int_{-1}^{1} \int_{-1}^{1} |\mathbf{J}(\xi, \eta)|\, f(\xi, \eta)\, \mathrm{d}\xi\, \mathrm{d}\eta$$

to evaluate this integral, carry out the integration over ξ through Gauss quadrature, in the same manner as in one dimension; this yields:

$$I = \int_{-1}^{1} \left(\int_{-1}^{1} |\mathbf{J}(\xi, \eta)|\, f(\xi, \eta)\, \mathrm{d}\xi \right) \mathrm{d}\eta \approx \int_{-1}^{1} \left(\sum_{i=1}^{n_{gp}} W_i\, |\mathbf{J}(\xi_i, \eta)|\, f(\xi_i, \eta) \right) \mathrm{d}\eta$$

finally, complete the integration by performing a Gauss quadrature over η:

$$I \approx \int_{-1}^{1} \left(\sum_{i=1}^{n_{gp}} W_i\, |\mathbf{J}(\xi_i, \eta)|\, f(\xi_i, \eta) \right) \mathrm{d}\eta \approx \sum_{j=1}^{n_{gp}} \left(\sum_{i=1}^{n_{gp}} W_j W_i\, \left|\mathbf{J}(\xi_i, \eta_j)\right|\, f(\xi_i, \eta_j) \right)$$

Gauss Quadrature in Two Dimensions

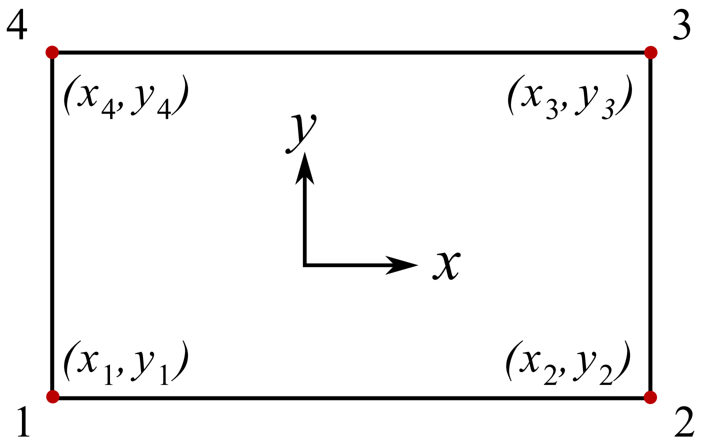
so the process for utilising Gauss quadrature is as follows:

- identify the function to be integrated
- determine how many Gauss points will be used and find the Gauss point locations and weights

- convert x (or whatever the variable of interest is) to ξ and get the Jacobian determinant
- evaluate the function at the Gauss points and multiply by the appropriate weights
- sum the product of the function evaluations and weight functions; this provides the approximated integral (exact for a polynomial given enough Gauss points)

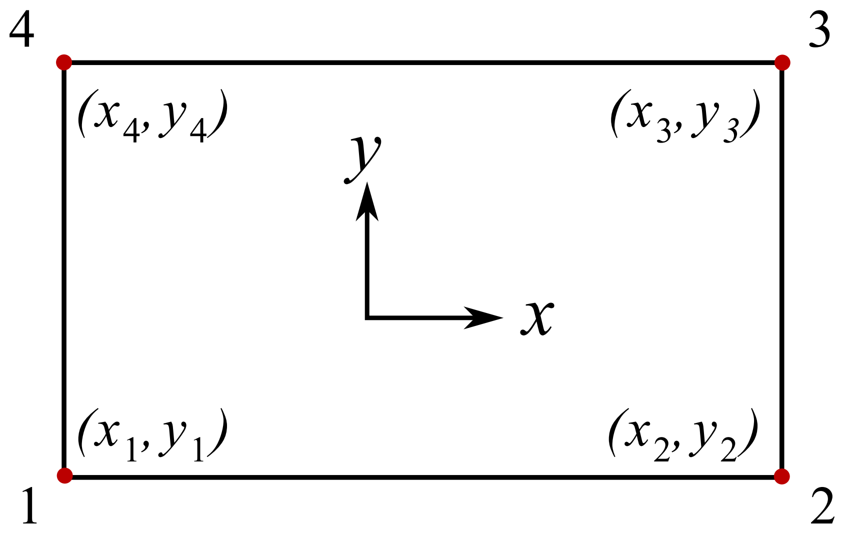
3.11 Two Dimensional Solid Quadrilateral Elements

Four-Node Rectangular Element



consider the four-node rectangular element above; again, the nodes are ordered anticlockwise

Four-Node Rectangular Element

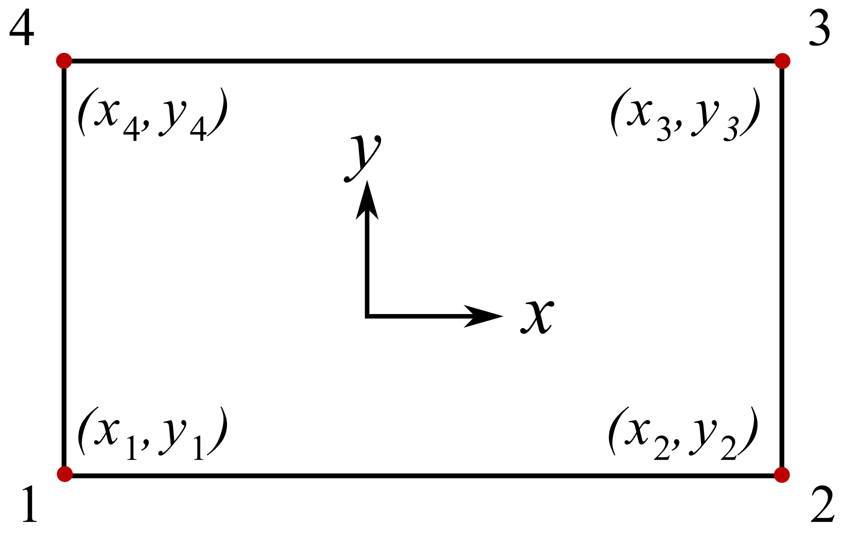


the three-node triangular element is the simplest element to derive and use in two dimensions, but it tends to be inaccurate because it models the strains as constant through the element

the weight functions for the triangular element have three constants: one associated with a constant term, and two associated with terms linear in x and y; this was consistent with the number of nodes in the element

a four-node element will need four constants; with what combination of x and y is the fourth constant associated?

Four-Node Rectangular Element



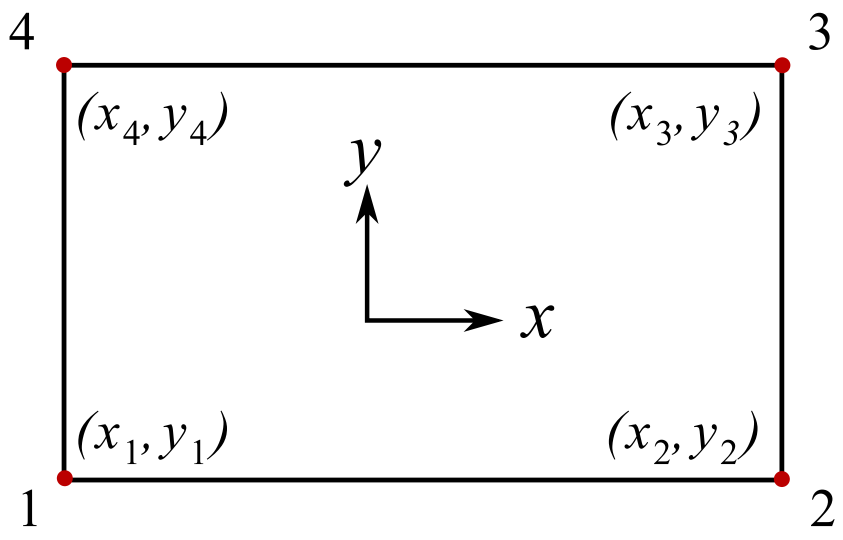
the answer to this comes from considering continuity between two elements

because the deformed geometry of each element is linked to the deflections at the nodes, the elements are automatically continuous at the nodes

however, continuity of displacements along the edges at which two elements meet is not necessarily ensured

consider the three-node triangle: the displacements along adjacent edges of separate elements were the same linear interpolations of the displacements at the nodes, which guaranteed continuity

Four-Node Rectangular Element



the same guarantee of continuity is necessary for rectangular elements, and to achieve this it is necessary to ensure linear interpolations along the sides of an element

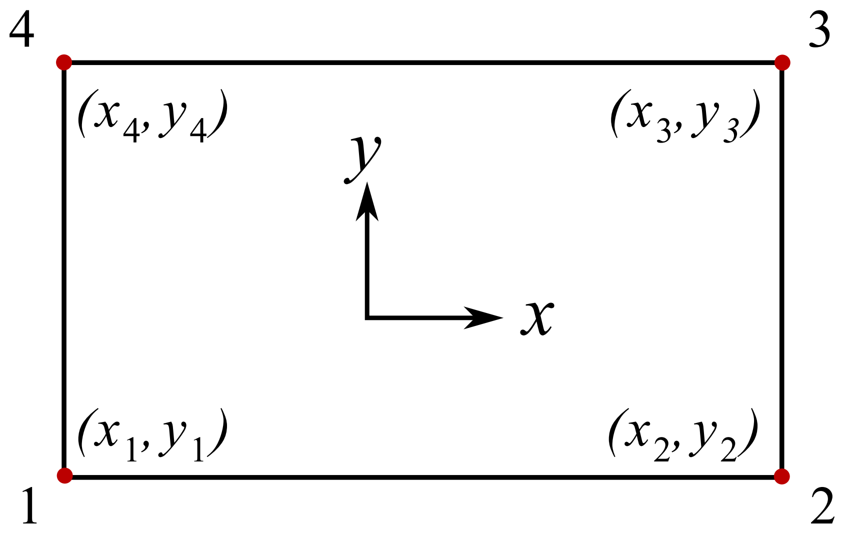
the shape function for the rectangular element will be:

$$u(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 y + \alpha_3 f(x, y),$$

but what is $f(x, y)$ that will maintain linear interpolations along the edges?

the solution will come from the third row of Pascal's triangle

Four-Node Rectangular Element



the term x^2 will result in displacements that are quadratic in x along the edges between 1 and 2 and between 3 and 4; a similar argument can be made for y^2

instead, using the term xy, one of x or y will be constant along any boundary (provided the boundaries are aligned with the coordinate axes) and hence the variation will be linear

this is the necessary term, and hence:

$$u(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 y + \alpha_3 xy$$

Four-Node Rectangular Element

the shape functions for the rectangular element are the next necessary component

the shape functions must obey the Kronecker delta rule: the displacement at one node has no effect on the displacement at other nodes; for example, $N_1(x_1, y_1) = 1$ while $N_1(x_2, y_2) = 0$

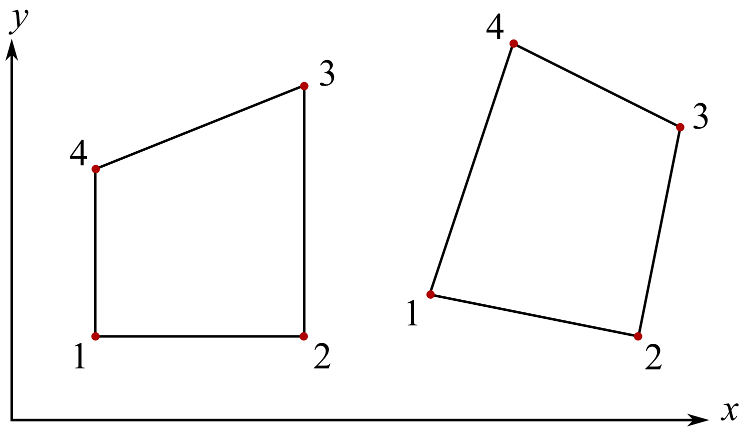
the four shape functions (which can be generated by multiplying the linear shape functions for a one-dimensional element in their various combinations) are:

$$\begin{aligned}
N_1 &= \frac{x - x_2}{x_1 - x_2}\frac{y - y_4}{y_1 - y_4} = \frac{1}{A}(x - x_2)(y - y_4) \\
N_2 &= \frac{x - x_1}{x_2 - x_1}\frac{y - y_3}{y_2 - y_3} = -\frac{1}{A}(x - x_1)(y - y_3) \\
N_3 &= \frac{x - x_4}{x_3 - x_4}\frac{y - y_2}{y_3 - y_2} = \frac{1}{A}(x - x_4)(y - y_2) \\
N_4 &= \frac{x - x_3}{x_4 - x_3}\frac{y - y_1}{y_4 - y_1} = -\frac{1}{A}(x - x_3)(y - y_1)
\end{aligned}$$

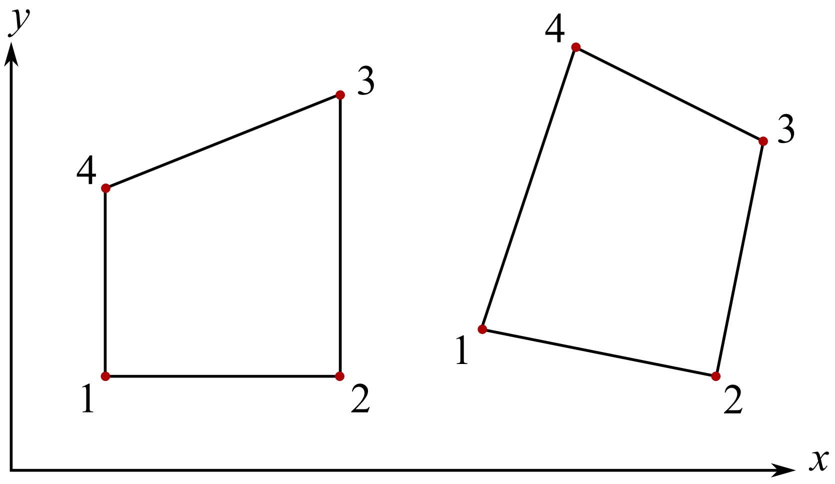
where A is the area of the element; note: $x_1 = x_4$, $x_2 = x_3$, $y_1 = y_2$, $y_3 = y_4$

Four-Node Rectangular Element

these shape functions will work for rectangles oriented with the coordinate axes; will it work for the elements below?



Isoparametric Element



the shape functions on the previous pages will not work for these elements because the element edges are not oriented parallel to the coordinate axes

consequently, the interpolations are not linear along the edges that are not parallel to the coordinate axes

hence, two nodes are not sufficient to ensure continuity of displacement along the boundary between two elements

this means that there might be jumps in displacement between elements

Isoparametric Element

the development of isoparametric elements is a key advance that makes finite element methods so widely applicable isoparametric elements permit arbitrarily shaped and oriented elements to be used in simulations; this allows complicated boundaries to be modelled accurately

recall Gauss quadrature, where the domain of integration was parameterised using ξ

isoparametric elements employ the same approach

using a one-dimensional element as an example, map the physical coordinate x to the parent coordinate ξ:

$$x = x_1 \frac{1-\xi}{2} + x_2 \frac{1+\xi}{2}$$

the parent coordinate ξ has domain $\xi \in [-1, 1]$ while x_1 and x_2 are the physical coordinates of the nodes

this provides a parameterisation of position within a one-dimensional element

Isoparametric Element

staying with the one-dimensional case, this also provides a parameterisation of the value of some property (displacement u, as an example) within the element based on the values at the nodes, using u_1 and u_2 the values of the displacement at nodes 1 and 2

take the interpolation for displacement $u(x)$ in terms of the physical coordinate x, and substitute the expression for x in terms of the parent coordinate ξ:

$$\begin{aligned} u &= u_1 \frac{x - x_2}{x_1 - x_2} + u_2 \frac{x - x_1}{x_2 - x_1} \\ &= u_1 \frac{x_1 (1-\xi) + x_2 (1+\xi) - 2x_2}{2(x_1 - x_2)} + u_2 \frac{x_1 (1-\xi) + x_2 (1+\xi) - 2x_1}{2(x_2 - x_1)} \\ &= u_1 \frac{1-\xi}{2} + u_2 \frac{1+\xi}{2} \end{aligned}$$

this is the same as the mapping from the physical coordinates to the parent coordinates and is crucial: the linear approximation of the solution function u is identical to the map from the physical coordinate to the parent coordinate

this is the essential feature of an isoparametric element

Isoparametric Element

moving to two dimensions, the parent element is a four-node two-unit by two-unit square, with nodal coordinates given by:

Node i	ξ_i	η_1
1	-1	-1
2	1	-1
3	1	1
4	-1	1

the physical position within an element can be treated like a property that is interpolated from the nodal values

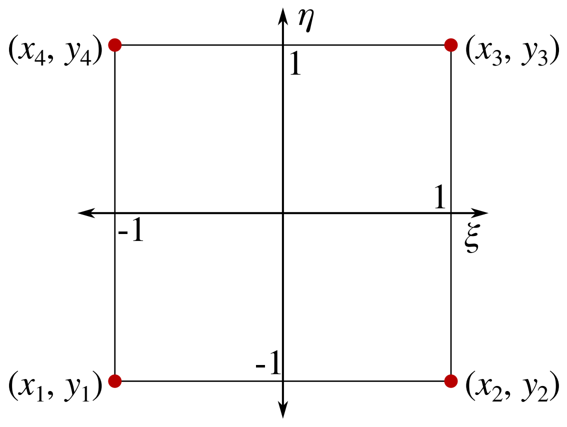
map the parent element to the physical element using the shape functions for the four-node rectangular element:

$$x(\xi,\eta) = \mathbf{N}^{4Q}\mathbf{x}; \qquad y(\xi,\eta) = \mathbf{N}^{4Q}\mathbf{y}$$

where:

$$\mathbf{x} = [x_1, x_2, x_3, x_4]^T \qquad \mathbf{y} = [y_1, y_2, y_3, y_4]^T$$

Isoparametric Element



Isoparametric Element

the shape functions for the parent element, $\mathbf{N}^{4Q}$, are given by:

$$N_1^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi_1\xi)(1+\eta_1\eta)$$

$$N_2^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi_2\xi)(1+\eta_2\eta)$$

$$N_3^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi_3\xi)(1+\eta_3\eta)$$

$$N_4^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi_4\xi)(1+\eta_4\eta)$$

these are obtained by replacing x and y in the shape functions for the four-node rectangular element with their parametric representations in terms of ξ and η

given values for $\left[\begin{array}{cc} \mathbf{x} & \mathbf{y} \end{array}\right]$, the physical nodal coordinates, the pair (x, y) can be determined by evaluating the shape functions at the desired parent coordinates (ξ, η) and multiplying by the physical nodal coordinates:

$$\left[\begin{array}{c} x \\ y \end{array}\right] = \mathbf{N}^{4Q}(\xi, \eta) \left[\begin{array}{cc} \mathbf{x} & \mathbf{y} \end{array}\right]$$

Isoparametric Element

the values of ξ_1, η_1 and the other nodal locations in the parent coordinate system are known:

$$N_1^{4Q}(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 - \eta)$$

$$N_2^{4Q}(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 - \eta)$$

$$N_3^{4Q}(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 + \eta)$$

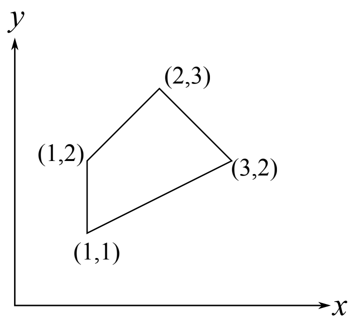
$$N_4^{4Q}(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 + \eta)$$

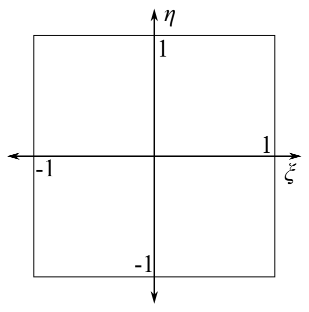
these are the shape functions for every four-noded quadrilateral element in the parent coordinate system

again, check that these shape functions obey the Kronecker delta rule

Isoparametric Element

take the element below on the left and map it onto the parent element on the right:





Isoparametric Element

generate the vectors of nodal positions:

$$\mathbf{x} = [1, 3, 2, 1]^T \qquad \mathbf{y} = [1, 2, 3, 2]^T$$

for any value of (ξ, η), where $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$, evaluate the shape functions and then multiply by the nodal coordinates:

$$x(\xi, \eta) = \mathbf{N}^{4Q}\mathbf{x}; \qquad y(\xi, \eta) = \mathbf{N}^{4Q}\mathbf{y}$$

this gives the values of (x, y) in the physical coordinate system for the known values of (ξ, η) in the parent coordinate system

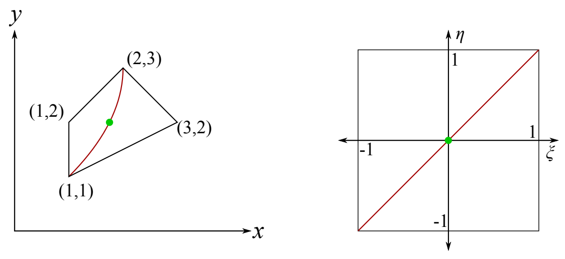
usually the mapping is from the parent element to the physical element; the mapping from physical to parent is typically only necessary for the nodes (the coordinates of the nodes of the parent element are known)

Isoparametric Element

in this example, $(\xi, \eta) = (0, 0)$ maps to $(x, y) = (1.75, 2)$; the values of the shape functions at this point are calculated by substituting $(\xi, \eta) = (0, 0)$ into the expressions for the shape functions to get:

$$\mathbf{N}^{4Q}(\xi = 0, \eta = 0) = [1/4, 1/4, 1/4, 1/4]$$

a straight line connecting nodes 1 and 3 in the parent element is mapped to a curved line in the physical element



straight lines on physical boundaries are mapped to straight lines in the parent element, but straight lines elsewhere in the physical element may be mapped to curved lines in the parent element

Isoparametric Element

the solution function is approximated by the same shape functions:

$$\mathbf{u}(\xi, \eta) \approx \mathbf{N}^{4Q}\mathbf{d}$$

and hence the element is isoparametric

the shape functions contain a constant term, a term linear in ξ, a term linear in η and a bilinear term $\xi\eta$

since the parent element is a square with sides aligned with the coordinate axes, linear interpolations are guaranteed along the element boundaries

isoparametric elements ensure continuity of displacement (or any other desired solution function) along the boundaries between elements regardless of the physical element shape

the next step is to calculate the stiffness matrices for isoparametric elements

Isoparametric Element

the element stiffness matrices require computation of integrals of the form:

$$\mathbf{K} = \int_{\Omega} \mathbf{H}^T \mathbf{D} \mathbf{H} \, \mathrm{d}\Omega$$

where **H** contains derivatives of the isoparametric shape functions

remember what **H** means: in physical coordinates, strain is given by the equation:

$$\boldsymbol{\epsilon} = \boldsymbol{\nabla}\mathbf{u} \approx \boldsymbol{\nabla}_S \mathbf{N}_I^{4Q}\mathbf{d} = \mathbf{H}\mathbf{d}$$

hence

$$\mathbf{H} = \boldsymbol{\nabla}_S \mathbf{N}^{4Q} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1^{4Q} & 0 & N_2^{4Q} & 0 & N_3^{4Q} & 0 & N_4^{4Q} & 0 \\ 0 & N_1^{4Q} & 0 & N_2^{4Q} & 0 & N_3^{4Q} & 0 & N_4^{4Q} \end{bmatrix}$$

Isoparametric Element

evaluating the vector-matrix product, the strain-displacement relation $\mathbf{H} = \boldsymbol{\nabla}_S \mathbf{N}_I^{4Q}$ is:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial N_1^{4Q}}{\partial x} & 0 & \frac{\partial N_2^{4Q}}{\partial x} & 0 & \frac{\partial N_3^{4Q}}{\partial x} & 0 & \frac{\partial N_4^{4Q}}{\partial x} & 0 \\ 0 & \frac{\partial N_1^{4Q}}{\partial y} & 0 & \frac{\partial N_2^{4Q}}{\partial y} & 0 & \frac{\partial N_3^{4Q}}{\partial y} & 0 & \frac{\partial N_4^{4Q}}{\partial y} \\ \frac{\partial N_1^{4Q}}{\partial y} & \frac{\partial N_1^{4Q}}{\partial x} & \frac{\partial N_2^{4Q}}{\partial y} & \frac{\partial N_2^{4Q}}{\partial x} & \frac{\partial N_3^{4Q}}{\partial y} & \frac{\partial N_3^{4Q}}{\partial x} & \frac{\partial N_4^{4Q}}{\partial y} & \frac{\partial N_4^{4Q}}{\partial x} \end{bmatrix}$$

the general rule is that the strain-displacement relation has dimensions (number of components of strain x number of degrees of freedom)

however, the shape functions are expressed in parent coordinates while the derivatives are with respect to the physical coordinates

so this must be transformed

Isoparametric Element

use the chain rule; for example, for the shape function for node 1:

$$\frac{\partial N_1^{4Q}}{\partial \xi} = \frac{\partial N_1^{4Q}}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_1^{4Q}}{\partial y}\frac{\partial y}{\partial \xi}$$

the collection of these relations can be written, where $I \in (1,2,3,4)$ as:

$$\begin{bmatrix} \dfrac{\partial N_I^{4Q}}{\partial \xi} \\ \\ \dfrac{\partial N_I^{4Q}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \dfrac{\partial N_I^{4Q}}{\partial x} \\ \\ \dfrac{\partial N_I^{4Q}}{\partial y} \end{bmatrix}$$

this uses the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}$$

Isoparametric Element

what are needed are the partials with respect to the physical coordinates, so invert the Jacobian matrix:

$$\begin{bmatrix} \dfrac{\partial N_I^{4Q}}{\partial x} \\ \\ \dfrac{\partial N_I^{4Q}}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dfrac{\partial N_I^{4Q}}{\partial \xi} \\ \\ \dfrac{\partial N_I^{4Q}}{\partial \eta} \end{bmatrix}$$

in vector notation, this becomes:

$$\mathbf{H} = \boldsymbol{\nabla}_S \mathbf{N}_I^{4Q} = \mathbf{J}^{-1}\mathbf{G}\mathbf{N}_I^{4Q}$$

where:

$$\mathbf{G} = \begin{bmatrix} \dfrac{\partial}{\partial \xi} \\ \\ \dfrac{\partial}{\partial \eta} \end{bmatrix}$$

is the gradient operator in the parent coordinate system

Isoparametric Element

the mapping between physical coordinates and parent coordinates is:

$$x(\xi,\eta) = \mathbf{N}^{4Q}\mathbf{x}; \qquad y(\xi,\eta) = \mathbf{N}^{4Q}\mathbf{y}$$

this can be used to calculate the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\[2ex] \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}$$

through the interpolations:

$$x(\xi,\eta) = \begin{bmatrix} N_1^{4Q} & N_2^{4Q} & N_3^{4Q} & N_4^{4Q} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$y(\xi,\eta) = \begin{bmatrix} N_1^{4Q} & N_2^{4Q} & N_3^{4Q} & N_4^{4Q} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Isoparametric Element

the nodal coordinates are not functions of the parent coordinates, but the shape functions are:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial N_1^{4Q}}{\partial \xi} & \dfrac{\partial N_2^{4Q}}{\partial \xi} & \dfrac{\partial N_3^{4Q}}{\partial \xi} & \dfrac{\partial N_4^{4Q}}{\partial \xi} \\[2ex] \dfrac{\partial N_1^{4Q}}{\partial \eta} & \dfrac{\partial N_2^{4Q}}{\partial \eta} & \dfrac{\partial N_3^{4Q}}{\partial \eta} & \dfrac{\partial N_4^{4Q}}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} \eta - 1 & 1 - \eta & 1 + \eta & -1 - \eta \\ \xi - 1 & -1 - \xi & 1 + \xi & 1 - \xi \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

converting to vector notation, this is:

$$\mathbf{J} = \mathbf{G}\mathbf{N}^{4Q}\begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix}$$

Isoparametric Element

this is a numerical process that enables the calculation of **J** for any location in the element

once **J** can be calculated, it can also be inverted to produce $\mathbf{J}^{-1}$, which is the mathematical object needed to calculate **H**

as an intermediate step, write $\mathbf{H}^*$ in matrix form as:

$$\mathbf{H}^* = \mathbf{J}^{-1}\mathbf{G}\mathbf{N}^{4Q}$$

providing:

$$\mathbf{H}^* = \begin{bmatrix} \dfrac{\partial N_1^{4Q}}{\partial x} & \dfrac{\partial N_2^{4Q}}{\partial x} & \dfrac{\partial N_3^{4Q}}{\partial x} & \dfrac{\partial N_4^{4Q}}{\partial x} \\ \dfrac{\partial N_1^{4Q}}{\partial y} & \dfrac{\partial N_1^{4Q}}{\partial y} & \dfrac{\partial N_3^{4Q}}{\partial y} & \dfrac{\partial N_4^{4Q}}{\partial y} \end{bmatrix}$$

these are the derivatives of the shape functions with respect to the physical coordinates, but are not yet the strain-displacement relation **H**

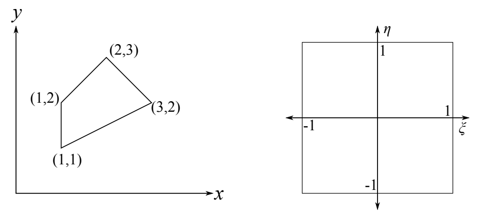
Isoparametric Element

finally, the terms in $\mathbf{H}^*$ must be distributed into **H**:

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial N_1^{4Q}}{\partial x} & 0 & \dfrac{\partial N_2^{4Q}}{\partial x} & 0 & \dfrac{\partial N_3^{4Q}}{\partial x} & 0 & \dfrac{\partial N_4^{4Q}}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1^{4Q}}{\partial y} & 0 & \dfrac{\partial N_2^{4Q}}{\partial y} & 0 & \dfrac{\partial N_3^{4Q}}{\partial y} & 0 & \dfrac{\partial N_4^{4Q}}{\partial y} \\ \dfrac{\partial N_1^{4Q}}{\partial y} & \dfrac{\partial N_1^{4Q}}{\partial x} & \dfrac{\partial N_2^{4Q}}{\partial y} & \dfrac{\partial N_2^{4Q}}{\partial x} & \dfrac{\partial N_3^{4Q}}{\partial y} & \dfrac{\partial N_3^{4Q}}{\partial x} & \dfrac{\partial N_4^{4Q}}{\partial y} & \dfrac{\partial N_4^{4Q}}{\partial x} \end{bmatrix}$$

Isoparametric Element

here is the example element from earlier



$$\mathbf{J} = \frac{1}{4}\begin{bmatrix} \eta - 1 & 1 - \eta & 1 + \eta & -1 - \eta \\ \xi - 1 & -1 - \xi & 1 + \xi & 1 - \xi \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 2 & 3 \\ 1 & 2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 3 - \eta & 2 \\ -1 - \xi & 2 \end{bmatrix}$$

$$|\mathbf{J}| = \frac{2\xi - 2\eta + 8}{16}$$

$$\mathbf{J}^{-1} = \frac{1}{2}\begin{bmatrix} 2 & -2 \\ 1+\xi & 3-\eta \end{bmatrix}$$

Isoparametric Element

the mapping from the physical element to the parent element must be unique and invertible; hence the determinant of the Jacobian matrix must be greater than zero for all x and y

this can be shown to be the case for all physical quadrilateral elements where the angles at the nodes are less than 180°

note that the shape functions do not depend upon the physical element coordinates (nor therefore size); this is the explanation for the superscripted $4Q$: they are universal functions associated with this element type

however, the Jacobian matrix does depend upon the physical element coordinates

Isoparametric Element - Force Vector

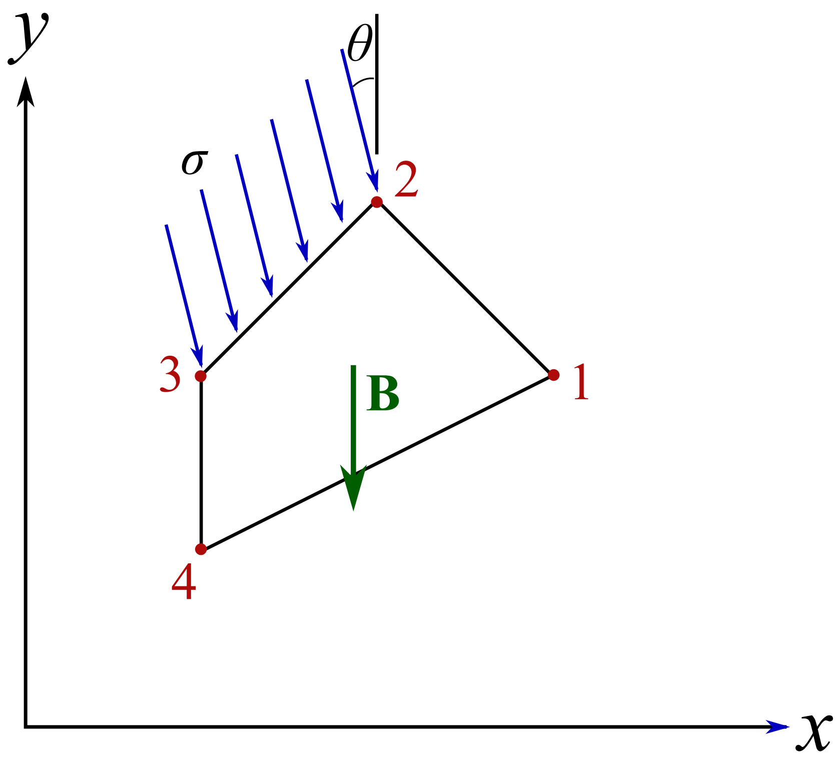
consider a linear quadrilateral isoparametric element loaded by a surface traction along the side between nodes 2 and 3 and by body force **B** distributed evenly throughout the element

the traction σ is not normal to the surface, but is aligned at angle θ to the vertical; note the orientation is indicated by the direction of the arrow

the traction $\overline{\mathbf{T}}$ has two components:

$$\bar{t}_x = \sigma \sin\theta$$

$$\bar{t}_y = -\sigma \cos\theta$$



Isoparametric Element - Force Vector

the general expression for the force vector due to the surface tractions is:

$$\mathbf{f} = \int_\Gamma \mathbf{N}^T \overline{\mathbf{T}} \, \mathrm{d}\Gamma$$

in this example, the traction is applied between nodes 2 and 3; along this boundary, $-1 \leq \eta \leq 1$ and $\xi = 1$

evaluating the integral provides:

$$\mathbf{f} = \int_\Gamma \mathbf{N}^T \overline{\mathbf{T}} \, \mathrm{d}\Gamma = t\,|\mathbf{J}| \int_{-1,\xi=1}^{1} \mathbf{N}^T \overline{\mathbf{T}} \, \mathrm{d}\eta = t\,|\mathbf{J}| \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{t}_x & \bar{t}_y \end{bmatrix} = t\frac{\ell}{2} \begin{bmatrix} 0 \\ 0 \\ \bar{t}_x \\ \bar{t}_y \\ \bar{t}_x \\ \bar{t}_y \\ 0 \\ 0 \end{bmatrix}$$

here t is the thickness of the element and with ℓ the length of the side between nodes 2 and 3, $|\mathbf{J}| = \ell/2$ is the Jacobian determinant of the transformation between the physical and the parent coordinate system along that side the effect is that half of the total force applied to the element side is applied to each node

Isoparametric Element - Force Vector

the general expression for the force vector due to body forces $\overline{\mathbf{B}}$ is:

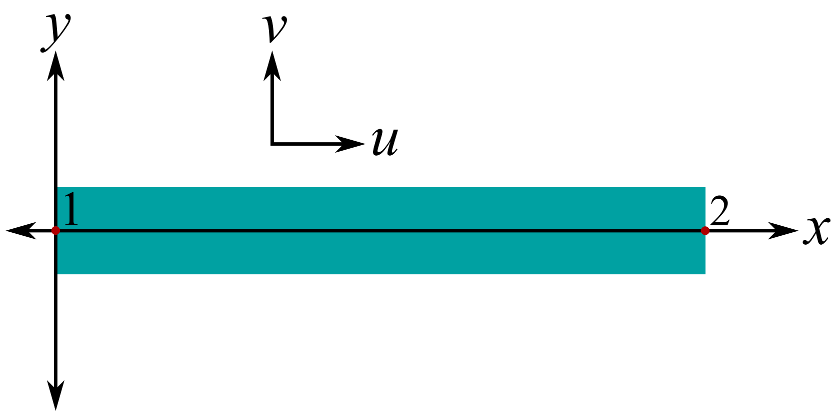
$$\mathbf{f} = \int_\Omega \mathbf{N}^T \overline{\mathbf{B}} \, \mathrm{d}\Omega$$

to evaluate this requires a Gauss quadrature over the area of the element, using n_{gp} Gauss points:

$$\mathbf{f} = \int_\Omega \mathbf{N}^T \overline{\mathbf{B}} \, \mathrm{d}\Omega = t \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} \left| \mathbf{J}(\xi_i, \eta_j) \right| w_i w_j \mathbf{N}^T (\xi_i, \eta_j) \overline{\mathbf{B}}$$

3.12 Beam Elements

Beam Elements



Beam Elements

another common finite element is the beam element, which can be used to model beams in bending

in order to capture bending behaviour accurately, higher order approximation is necessary, so this is an example of a higher order element

number the nodes at the end of the beam element 1 and 2

this derivation will use Bernoulli beam theory (Timoshenko beam theory could alternately be used if the additional complication is not prohibitive)

remember the main assumptions of Bernoulli beam theory:

1. the beam lies along the x-axis

2. the deflections of interest, v, are normal to the beam, in the y direction
3. sections normal to the axis of the beam remain normal to the axis of the beam after bending; for symmetric beams, the axis (and the neutral axis of bending) is the mid-line

Beam Elements

the displacements of a point on the beam are given by u and v, aligned with the x and y axes

the normality assumption determines the u component of displacement through the thickness of the beam by:

$$u = -y \sin\theta(x)$$

where $\theta(x)$ is the rotation of the axis of the beam after deflection at location x

if the deflections are small, then the angle $\theta(x)$ is small and $\sin\theta \approx \theta$, making the angle of rotation correspond to the slope of the mid-line:

$$\theta(x) = \frac{\partial v}{\partial x}$$

combining these two equations gives:

$$u = -y\frac{\partial v}{\partial x}$$

Beam Elements

the expression for axial strain is:

$$\epsilon_{xx} = \frac{\partial u}{\partial x} = -y\frac{\partial^2 v}{\partial x^2}$$

assume that v is only a function of x

consider a more general case when the mid-line is also stretched or compressed; by superposition:

$$u = u^M - y\frac{\partial v}{\partial x}$$

where u^M is the displacement at the mid-line

this generates three expressions for strain:

$$\epsilon_{xx} = \frac{\partial u^M}{\partial x} - y\frac{\partial^2 v}{\partial x^2} \qquad \epsilon_{yy} = \frac{\partial v}{\partial y} = 0 \qquad \epsilon_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

Beam Elements

assume that Hookean linear elasticity holds; the stress-strain law is:

$$\sigma_{xx} = E\epsilon_{xx} = E\left(\frac{\mathrm{d}u^M}{\mathrm{d}x} - y\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\right)$$

note that these are total derivatives, since both u^M and v are functions of x only

the stresses on the beam section can be related directly to the moment by integrating:

$$M = -\int_A y\sigma_{xx}\,\mathrm{d}A = -\int_A yE\left(\frac{\mathrm{d}u^M}{\mathrm{d}x} - y\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\right)\mathrm{d}A = \int_A Ey^2\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\,\mathrm{d}A$$

recall that:

$$I = \int_A y^2\,\mathrm{d}A$$

and therefore:

$$M = EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} = EI\kappa$$

which is a constitutive law

Beam Elements

the transverse equilibrium statement for a beam element is:

$$\left(V + \frac{\mathrm{d}V}{\mathrm{d}x}\,\mathrm{d}x\right) - V + p\left(\frac{\mathrm{d}x}{2}\right)\mathrm{d}x = 0$$

where V is the shear force and p is the distributed force (evaluated at the centre of the differential element)

in the limit, this gives:

$$\frac{\mathrm{d}V}{\mathrm{d}x} + p = 0$$

moment equilibrium states that:

$$\left(M + \frac{\mathrm{d}M}{\mathrm{d}x}\,\mathrm{d}x\right) - M + \left(V + \frac{\mathrm{d}V}{\mathrm{d}x}\,\mathrm{d}x\right)\mathrm{d}x + \frac{1}{2}\,\mathrm{d}x^2 p\left(x + \frac{\mathrm{d}x}{2}\right) = 0$$

which, in the limit, gives:

$$\frac{\mathrm{d}M}{\mathrm{d}x} + V = 0$$

Beam Elements

amalgamating all of these equations produces:

$$\frac{\mathrm{d}^2 M}{\mathrm{d}x^2} - p = 0,$$

which, along with $M = EI\kappa$, is a strong form of the beam equations combining these generates the basic expression of Bernoulli beam theory:

$$EI\frac{\mathrm{d}^4 v}{\mathrm{d}x^4} - p = 0$$

the above is a fourth order ordinary differential equation for the vertical displacement of the mid-line of the beam; note that this is of higher order than the equations for the one-dimensional or two-dimensional finite elements

Beam Elements - Boundary Conditions

boundary conditions are needed to solve this equation; there are several possibilities:

1. $v = \overline{v}$ on Γ_u (essential)
2. $\frac{\mathrm{d}v}{\mathrm{d}x} = -\overline{\theta}$ on Γ_θ (essential)
3. $EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} = \overline{M}$ on Γ_M (natural)
4. $-EI\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} = \overline{V}$ on Γ_V (natural)

these can be combined in several ways:

1. at a free end with an applied load: $V = \overline{V}$, $M = \overline{M}$
2. at a simple support: $v = 0$, $M = 0$
3. at a clamped support: $v = 0$, $\frac{\mathrm{d}v}{\mathrm{d}x} = 0$

an important consideration is that two conditions that are work conjugate cannot be specified at a single point on the boundary

Beam Elements - Weak Form

to find the weak form for the beam element, multiply the equilibrium equation and the natural boundary conditions by the weight function or its derivative and then integrate the equilibrium equation:

$$\int_\Omega \delta w\left(\frac{\mathrm{d}^2 M}{\mathrm{d}x^2} - p\right)\mathrm{d}x = 0$$

$$\delta w\left(-EI\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \overline{V}\right)\Bigg|_{\Gamma_V} = 0$$

$$\frac{\mathrm{d}\delta w}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} - \overline{M}\right)\Bigg|_{\Gamma_M} = 0$$

these will lead to expressions for the weak form of the equilibrium equations for the beam element

Beam Elements - Weak Form

integrate by parts:

$$\int_\Omega \delta w \frac{\mathrm{d}^2 M}{\mathrm{d}x^2}\,\mathrm{d}x = \left(\delta w \frac{\mathrm{d}M}{\mathrm{d}x}\right)\bigg|_{\Gamma_V} - \int_\Omega \frac{\mathrm{d}\delta w}{\mathrm{d}x}\frac{\mathrm{d}M}{\mathrm{d}x}\,\mathrm{d}x = \left(-\delta w \overline{V}\right)\Big|_{\Gamma_V} - \int_\Omega \frac{\mathrm{d}\delta w}{\mathrm{d}x}\frac{\mathrm{d}M}{\mathrm{d}x}\,\mathrm{d}x$$

this uses the divergence theorem to move from a volume integral over the body (Ω) to a surface integral over the boundary (Γ)

then integrate the second term above by parts:

$$\int_\Omega \frac{\mathrm{d}\delta w}{\mathrm{d}x}\frac{\mathrm{d}M}{\mathrm{d}x}\,\mathrm{d}x = \left(\frac{\mathrm{d}\delta w}{\mathrm{d}x}\overline{M}\right)\bigg|_{\Gamma_M} - \int_\Omega \frac{\mathrm{d}^2\delta w}{\mathrm{d}x^2} M\,\mathrm{d}x$$

Beam Elements - Weak Form

combine all of the above results and substitute back into the original weak form of the equilibrium equation to get:

$$\int_\Omega \frac{\mathrm{d}^2\delta w}{\mathrm{d}x^2} M\,\mathrm{d}x = \int_\Omega \delta w p\,\mathrm{d}x + \left(\frac{\delta w}{\delta x}\overline{M}\right)\bigg|_{\Gamma_M} + \left(-\delta w \overline{V}\right)\Big|_{\Gamma_V}$$

put Hooke's Law into this equation to produce:

$$\int_\Omega \frac{\mathrm{d}^2\delta w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\,\mathrm{d}x = \int_\Omega \delta w p\,\mathrm{d}x + \left(\frac{\delta w}{\delta x}\overline{M}\right)\bigg|_{\Gamma_M} + \left(-\delta w \overline{V}\right)\Big|_{\Gamma_V}$$

what impact does the nature of this equation have on the allowable functions we can use for δw and the approximation function?

Beam Elements - Discretization

this equation has higher continuity requirements than for the one-dimensional element: beam elements require both displacement continuity and slope continuity at the boundaries of the element and hence the approximation functions must be C^1

this will require additional degrees of freedom for the element: two displacement degrees of freedom (four if axial compression or stretching is allowed; here it is not) and two rotational degrees of freedom, one of each at each node:

$$\mathbf{d} = [v_1, \theta_1, v_2, \theta_2]^T$$

the nodal forces are work conjugate to the nodal displacements:

$$\mathbf{f} = [F_1, M_1, F_2, M_2]^T$$

Beam Elements - Discretization

the shape functions for the beam element weight function and solution function are:

$$N_1^v = \frac{1}{4}(1-\xi)^2(2+\xi)$$

$$N_1^\theta = \frac{\ell}{8}(1-\xi)^2(1+\xi)$$

$$N_2^v = \frac{1}{4}(1+\xi)^2(2-\xi)$$

$$N_2^\theta = \frac{\ell}{8}(1+\xi)^2(\xi-1)$$

in a local coordinate system with the origin at one end of the element with position parameterised by ξ, which is:

$$\xi = \frac{2x}{\ell} - 1$$

where ℓ is the length of the element and $-1 \leq \xi \leq 1$

check that the shape functions obey the Kronecker delta properties and note that this is again similar to mapping in Gauss quadrature

Beam Elements - Discretization

the shape functions are used to interpolate both the solution and the weight function and the derivatives of both from the nodal values **d** and **w**:

$$v \approx \mathbf{Nd} \quad \delta w \approx \mathbf{Nw}$$

the derivatives are with respect to x so use the Jacobian determinant to change variables:

$$\frac{\mathrm{d}}{\mathrm{d}x} = \frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}\xi} = \frac{2}{\ell}\frac{\mathrm{d}}{\mathrm{d}\xi}$$

the second derivative of the shape function is:

$$\frac{\mathrm{d}^2\mathbf{N}}{\mathrm{d}x^2} = \frac{1}{\ell}\begin{bmatrix} \frac{6\xi}{\ell} & 3\xi - 1 & \frac{-6\xi}{\ell} & 3\xi + 1 \end{bmatrix} = \mathbf{B}$$

hence:

$$\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} \approx \mathbf{Bd} \quad \frac{\mathrm{d}^2\delta w}{\mathrm{d}x^2} \approx \mathbf{Bw}$$

Beam Elements - Discrete Equations

in matrix form, the discrete equations are the same as for other elements:

$$\mathbf{Kd} = \mathbf{f} + \mathbf{r}$$

where **r** is the residual, which contains both the reaction forces and small errors due to approximation and discretization the stiffness matrix for the element is:

$$\mathbf{K} = \int_\Omega \mathbf{B}^T EI\mathbf{B}\,\mathrm{d}x$$

and the external force vector is:

$$\mathbf{f} = \int_\Omega \mathbf{N}^T p\,\mathrm{d}x + \left(\mathbf{N}^T\overline{V}\right)\Big|_{\Gamma_V} + \left(\frac{\mathrm{d}\mathbf{N}^T}{\mathrm{d}x}\overline{M}\right)\Big|_{\Gamma_M}$$

Beam Elements - Discrete Equations

if the stiffness E and second moment of area I are constant through the element, the stiffness matrix is given by:

$$\mathbf{K} = \frac{EI}{\ell^3}\begin{bmatrix} 12 & 6\ell & -12 & 6\ell \\ 6\ell & 4\ell^2 & -6\ell & 2\ell^2 \\ -12 & -6\ell & 12 & -6\ell \\ 6\ell & 2\ell^2 & -6\ell & 4\ell^2 \end{bmatrix}$$

and the nodal forces associated with a constant applied distributed load p:

$$\mathbf{f} = \int_\Omega \mathbf{N}^T p\,\mathrm{d}x = \int_0^\ell \mathbf{N}^T p\,\mathrm{d}x = \frac{p\ell}{12}\begin{bmatrix} 6 \\ \ell \\ 6 \\ -\ell \end{bmatrix}$$

3.13 Solution Methods

System Equations

the individual element stiffness matrices are combined into a global stiffness matrix through a systematic process

this process is described in the finite element literature as “scatter” and “gather”

distinguish between element matrices, which have no superscript, and global matrices, which have the superscript g

the element stiffness relation is:

$$\mathbf{f} = \mathbf{K}\mathbf{d}$$

whereas the global relation:

$$\mathbf{f}^g + \mathbf{r}^g = \mathbf{K}^g\mathbf{d}^g$$

is the equation that eventually needs solution

here, $\mathbf{r}^g$ is a reaction - residual term that contains the reaction forces and any small numerical inaccuracies due to the approximations

System Equations

take a one-dimensional system consisting of two two-noded elements

for element 1, the element nodal displacement is:

$$\mathbf{d} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

while the system global nodal displacement is given by:

$$\mathbf{d}^g = \begin{bmatrix} u_1^g \\ u_2^g \\ u_3^g \end{bmatrix}$$

the relationship between the element and global displacements is:

$$\mathbf{d} = \mathbf{L}\mathbf{d}^g$$

where:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

System Equations

the matrix **L** is called the "gather" matrix because it collects the element terms from the global matrix

the gather matrix is different for each element, and relates the local element node numbering to the global node numbering

for the example, the gather matrix for element 2 is:

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

these matrices are very important because they connect not only the element displacements to the global displacements, but also the element stiffness matrices and element force vectors to their global counterparts

note, however, that actually constructing and employing these matrices in Matlab leads to very slow code

System Equations

for a single element:

$$\mathbf{K}\mathbf{L}\mathbf{d}^g = \mathbf{f}$$

examine the force vector; it can be transformed in a similar manner to the displacement vector:

$$\mathbf{f}^g = \begin{bmatrix} f_1^g \\ f_2^g \\ f_3^g \end{bmatrix}; \quad \mathbf{L}^T\mathbf{f} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

where $\mathbf{L}^T$ is called the “scatter” matrix

the global force vector can be assembled from the element force vectors:

$$\sum_{elements} \mathbf{L}^T\mathbf{f} = \mathbf{f}^g + \mathbf{r}^g$$

System Equations

use this to modify the global equation:

$$\sum_{elements} \left(\mathbf{L}^T\mathbf{K}\mathbf{L}\right)\mathbf{d}^g = \sum_{elements} \left(\mathbf{L}^T\mathbf{f}\right) = \mathbf{f}^g + \mathbf{r}^g$$

the key part of this is the stiffness matrix: the global stiffness matrix is:

$$\mathbf{K}^g = \sum_{elements} \mathbf{L}^T\mathbf{K}\mathbf{L}$$

this provides the final form to solve for the complete finite element system:

$$\mathbf{K}^g\mathbf{d}^g = \mathbf{f}^g + \mathbf{r}^g$$

Partitioned System

the final set of system equations typically has the form:

$$\begin{bmatrix} \vdots \\ f_7 \\ f_8 \end{bmatrix} = \left[\begin{array}{cc|cc} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \hline \vdots & \vdots & K_{77} & K_{78} \\ \vdots & \vdots & K_{87} & K_{88} \end{array}\right] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ d_7 \\ d_8 \end{bmatrix}$$

where many of the elements of the vector **d** are known to be zero

in that case, it is only necessary to solve the partitioned system:

$$\begin{bmatrix} P_7 \\ P_8 \end{bmatrix} = \begin{bmatrix} K_{77} & K_{78} \\ K_{87} & K_{88} \end{bmatrix} \begin{bmatrix} u_7 \\ u_8 \end{bmatrix}$$

Partitioned System

in general, it is possible to remove the elements of the displacement and nodal force vectors, and the corresponding rows and columns of the stiffness matrix, for elements of the displacement vector known to be zero from the boundary conditions

typically, the displacements that are zero due to the boundary conditions are not quite so easily arranged into a single group as in this example

clever node numbering can often achieve this, but in general it is necessary to use another algorithm to remove the matrix rows and columns that are unnecessary; note that, if element *i* in the displacement vector is zero because of the boundary conditions, in MATLAB the command:

```
K(i,:)  = []; K(:,i) = [];
```

will remove the relevant elements from the stiffness matrix **K**

the same operation must be carried out on the vectors of displacements and nodal forces as well

Element Load Vector

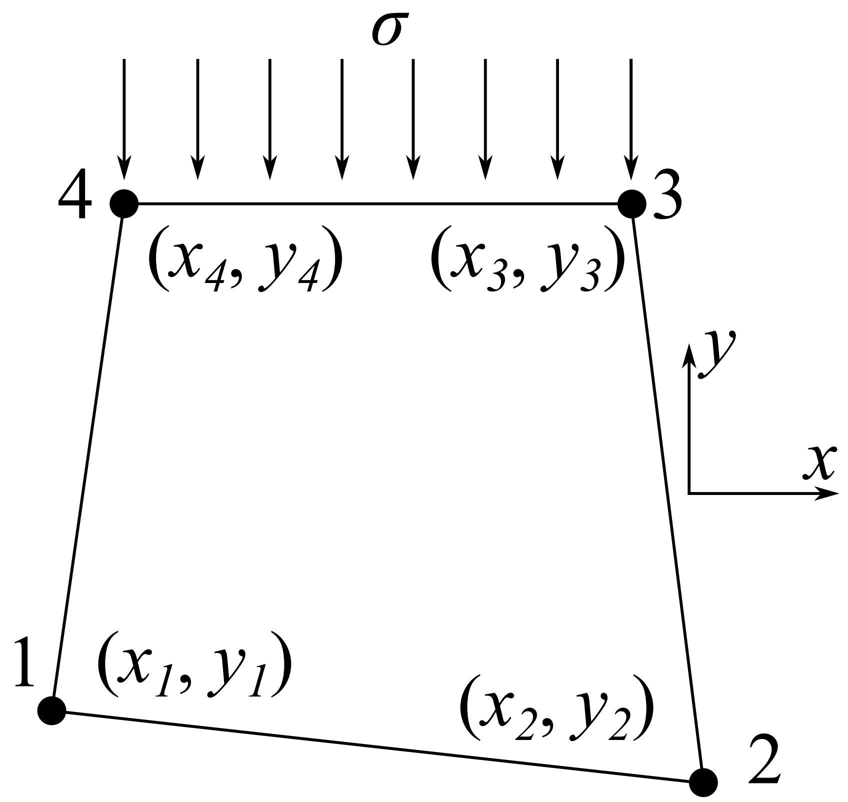
for each element, there is a load vector with size equal to the number of degrees of freedom in the element

for a two-dimensional four-node quadrilateral element with a vector-valued solution (for example, in stress analysis), there are eight degrees of freedom for each element, two at each node

this load vector represents the distribution of external forces acting on the element, distributed amongst the nodes; forces can only act at the nodes in the approximated, discretised system

the external forces can be either surface tractions or body forces

Element Load Vector



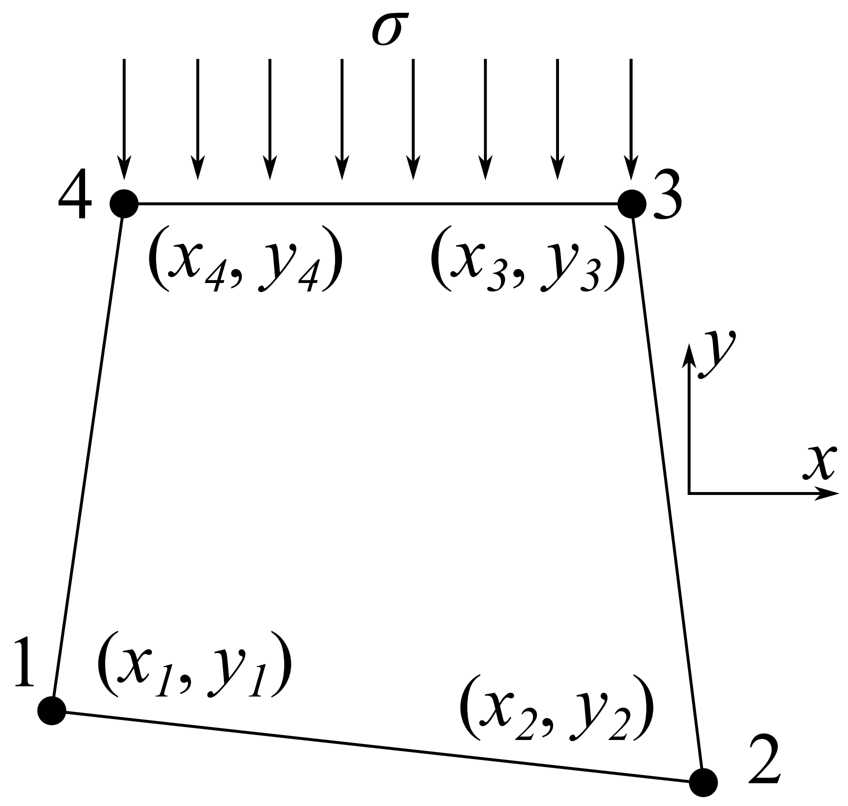
distributed loads, as seen to the left, must allocated to the element nodes

typically, if *b* is the thickness of the element, then the loads in the *y*-direction applied to nodes 3 and 4 would be equal to:

$$f_y = b\frac{(x_3 - x_4)}{2}\sigma$$

note that these are forces, not stresses or tractions

Element Load Vector

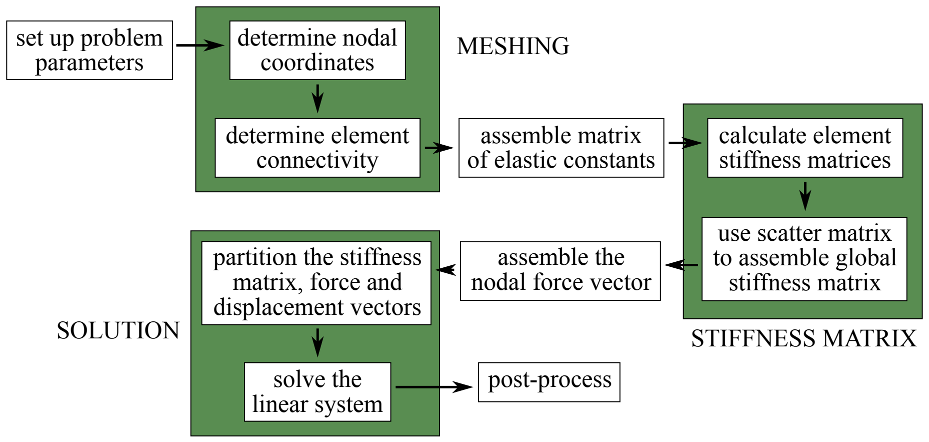


as a consequence, the nodal force vector for this element would be:

$$f = \begin{bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ f_y \\ 0 \\ f_y \end{bmatrix}$$

here, f_{1x} (for example) is the nodal force at node 1 in the x-direction

Structure of a Finite Element Code


set up problem parameters
determine nodal coordinates
MESHING
determine element connectivity
assemble matrix of elastic constants
calculate element stiffness matrices
use scatter matrix to assemble global stiffness matrix
STIFFNESS MATRIX
assemble the nodal force vector
partition the stiffness matrix, force and displacement vectors
SOLUTION
solve the linear system
post-process

Meshing

not much has been said about meshing; a good mesh is critical to the solution of a finite element problem

a “good” mesh is one that is sufficiently fine in areas where the strains are changing rapidly to capture the details of the material behaviour, coarse in regions where the strains are relatively constant, and with elements that are fairly regularly shaped: as close to square as possible for quadrilateral elements

to create a mesh, determine the locations of the nodes that will serve as the boundaries of the elements; it is generally a good idea to divide the area of interest into regions based upon an intuitive understanding of whether the strain gradients will be large or small

ideally, this could be done in an automated fashion but it usually requires substantial user intervention even with commercial packages

Mesh Convergence

it is always necessary to check that the mesh used for a finite element calculation achieves convergence

this means that increasing the number of elements does not have a significant effect on the final result

typically a convergence study would be performed by taking an existing finite element solution and comparing the solution to a new solution generated with four times (or sixteen times) as many elements

if there is minimal change between the two solutions (with minimal defined by the user depending upon the circumstances, but often by requiring that no displacement changes more than 0.5%), then the solution is considered to have converged with respect to the mesh

Meshing

in general, it is a good idea to store the coordinates of the nodes in separate vectors, that is:

$$\mathbf{x}(i) \quad \text{and} \quad \mathbf{y}(i)$$

where, for example, `y(10)` is the y-coordinate of node 10

the next step is to determine the nodes corresponding to the element vertices; remember to order them anticlockwise

these data are typically stored in matrix format, in a matrix **e** which has dimensions (number of elements x number of nodes per element); thus, in MATLAB,

```
x(e(12,2))
```

provides the x-coordinate of the second node of element 12

Meshing

most commercial finite element packages take care of the meshing of a finite element problem based upon geometric input and some guidance from the user on where elements should be finer or coarser

there are specialized meshing packages that can generate meshes from user-defined geometry and input, and are also able to work with output from CAD programs more easily than can finite element packages themselves

even with automated meshing, it is incumbent upon the user to check the mesh very carefully for problems; poor meshes can lead to very poor results with concomitant consequences for engineering decision-making

Stiffness Matrix

the calculation of the stiffness matrix is the heart of the finite element method, but actually takes much less code than meshing or post-processing do

for each element:

1. determine the element nodal coordinates, node numbers and degree of freedom numbers
2. assemble the scatter matrix
3. for each Gauss point in an element, the usual process is:
   (a) calculate the shape functions for that Gauss point
   (b) calculate the derivatives of the shape function at the Gauss point
   (c) calculate the Jacobian matrix and the Jacobian determinant
   (d) calculate the derivative matrix **H** for the Gauss point
   (e) calculate the element stiffness contribution for the Gauss point:

$$\mathbf{K_G} = W_i W_j \mathbf{H}^T(\xi_i, \eta_j)\mathbf{D}\mathbf{H}(\xi_i, \eta_j)\left|\mathbf{J}(\xi_i, \eta_j)\right|$$

(f) scatter the Gauss point stiffness contribution into the global matrix

it is often helpful to retain some of the intermediate calculations for use later during post-processing

Post-Processing

there is no limit to the amount of code that can be written to do post-processing of finite element results

the solution of the matrix equation is the vector of displacements of the nodes (the displacements known to be zero can be added back at this point)

using this vector, the new nodal locations can be calculated; these can be used to generate a deformed mesh, which is often shown (with the deformations magnified) in conjunction with the undeformed mesh

with the complete (non-partitioned) displacement vector, the complete matrix can be used to calculate the complete nodal force vector; subtracting the external force vector gives the vector of reaction forces plus the residual (which contains any error in the linear solution)

Post-Processing

using the nodal displacements and the **H** matrices, the strains can be calculated at each *Gauss point*; note that the strains at the nodes have not been calculated directly

strains elsewhere can be interpolated from the strains at the Gauss points

however, the Gauss points are typically not arranged in any orderly fashion if the mesh is non-uniform or contains elements with faces not parallel to the coordinate axes

this requires additional work to accomplish the interpolations

once the strains are known, the matrix of elastic constants can be used to determine the stresses, again at the Gauss points

the same comment about interpolation applies here

Post-Processing

other measures of stress and strain can be calculated; often, local principal stresses or von Mises stresses are of interest

often when looking at stress and strain, it is helpful to plot contours of stress and strain; Matlab has powerful tools to do this, but there are some restrictions on the form of the input matrices that are allowable

other quantities that may be of interest include vectors of displacements or contours of strain energy density