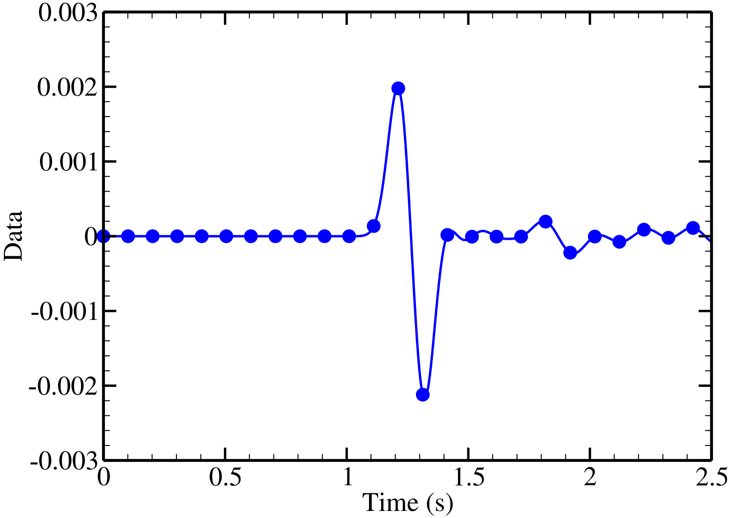
Master of Science on Computational Science

Institute of Computational Science

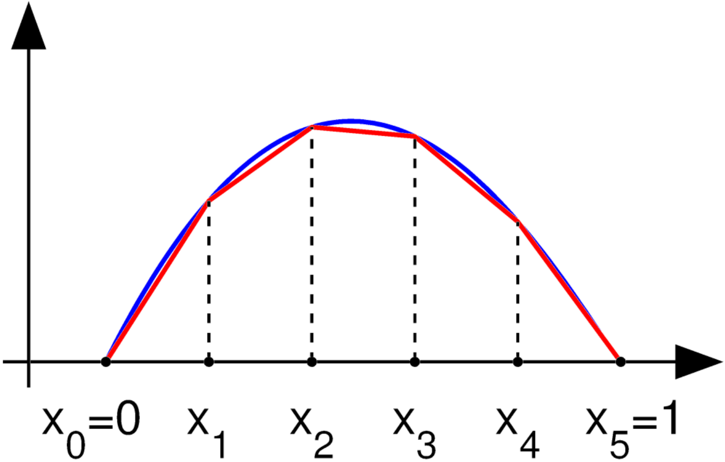
prof. Dr. Rolf Krause & Dr. Drosos Kourounis

24 Sep 2015, LAB2

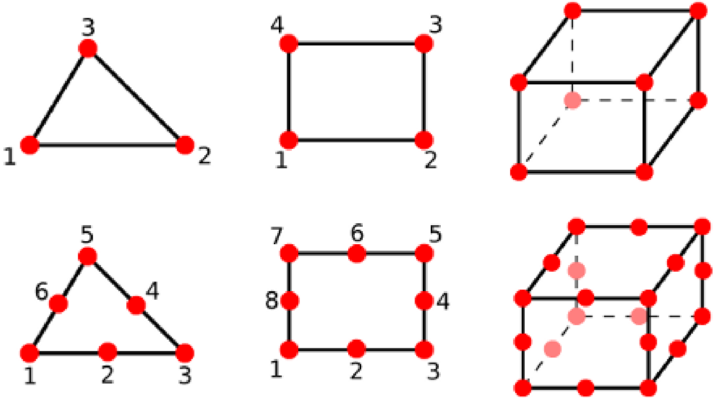
Interpolation in one dimension (1D)



Finite Elements in one dimension (1D)



2D and 3D Finite Elements



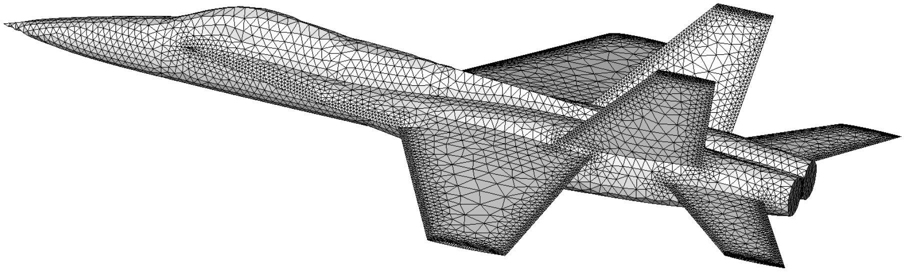
Triangle element with 3 and 6 nodes

Rectangle element with 4 and 8 nodes

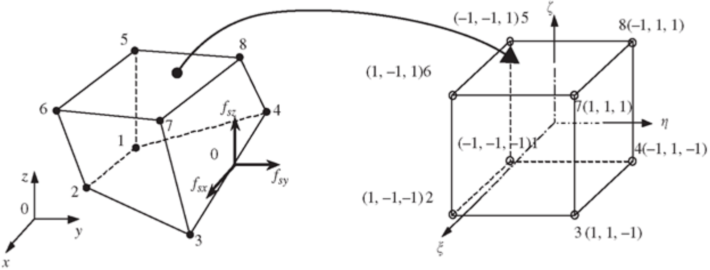
Box element (for 3D) with 8 and 20 nodes

Sample of some simple element shapes and standard node placement. By convention nodes are numbered anti-clockwise.

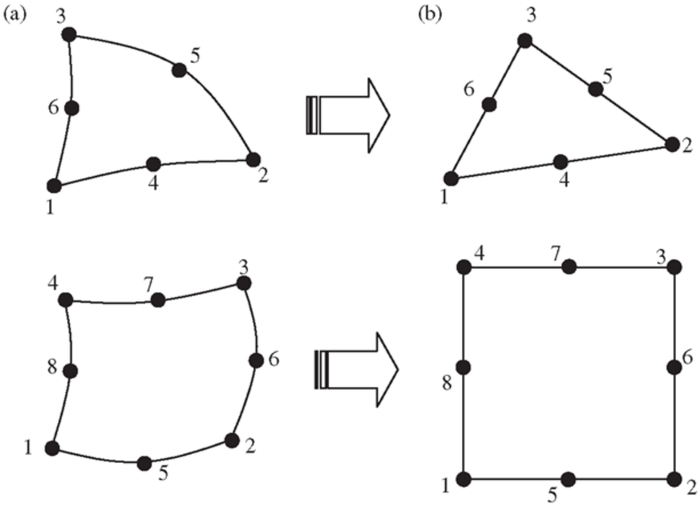
The 3D case can be very challenging



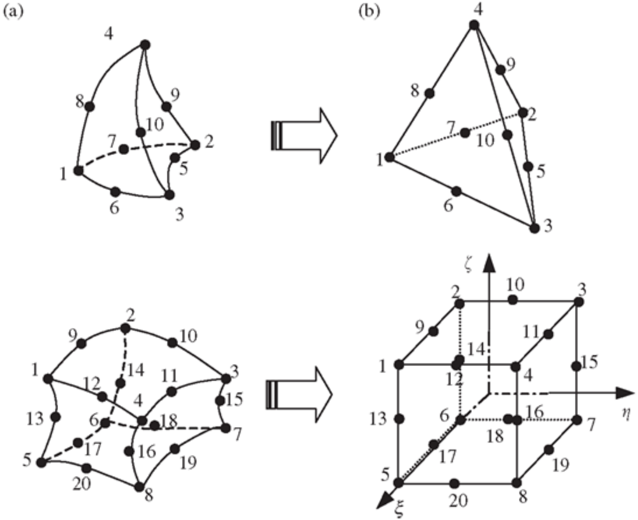
Mapping to the reference element 3D



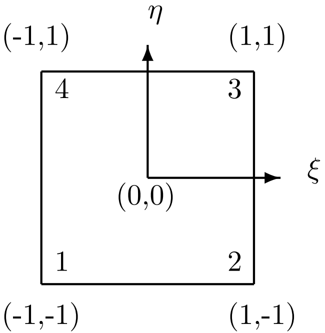
Mapping to the reference in 2D



Mapping to the reference in 3D



The 2D quadrilateral element



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

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

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

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

$$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$$

The 2D quadrilateral element ...

Any unknown field u as well as the coordinates (x, y) may be expressed as functions of (ξ, η) as

$$u = N_1 u_1 + N_2 u_2 + N_3 u_3 + N_4 u_4$$
$$x = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4$$
$$y = N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4.$$

Now suppose $f = f(x, y) = f(x(\xi, \eta), y(\xi, \eta))$. Using the chain rule of differentiation, we have

$$\frac{\partial f}{\partial \xi} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial \xi}$$
$$\frac{\partial f}{\partial \eta} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial \eta}$$

The 2D quadrilateral element ...

$$\begin{pmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{pmatrix} = J \begin{pmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{pmatrix}, \quad J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix}$$

$$\begin{pmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{pmatrix} = J^{-1} \begin{pmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{pmatrix}, \quad dxdy = |J|\, d\xi d\eta$$

The 2D quadrilateral element ...

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}$$

$$\begin{aligned}
J_{11} &= \frac{1}{4}\left(-(1-\eta)x_1 + (1-\eta)x_2 + (1+\eta)x_3 - (1+\eta)x_4\right) \\
J_{12} &= \frac{1}{4}\left(-(1-\eta)y_1 + (1-\eta)y_2 + (1+\eta)y_3 - (1+\eta)y_4\right) \\
J_{21} &= \frac{1}{4}\left(-(1-\xi)x_1 - (1+\xi)x_2 + (1+\xi)x_3 + (1-\xi)x_4\right) \\
J_{22} &= \frac{1}{4}\left(-(1-\xi)y_1 - (1+\xi)y_2 + (1+\xi)y_3 + (1-\xi)y_4\right)
\end{aligned}$$

2D integrals

Gaussian quadrature in 1D: n points can integrate exactly a polynomial of degree equal to $2n-1$

$$I = \int_{-1}^{1} f(\xi) d\xi$$

$$I \approx \sum_{k=1}^{n} w_k f(\xi_k)$$

Gaussian quadrature in 2D

$$I = \int_{-1}^{1} \int_{-1}^{1} f(\xi, \eta) d\xi \; d\eta$$

$$I \approx \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} w_{k_1} w_{k_2} f(\xi_{k_1}, \eta_{k_2})$$

2D integrals ...

Integrating the Laplacian on the reference element

$$I = \int_{\Omega} \nabla N_i(x,y) \cdot \nabla N_j(x,y) d\Omega$$

$$I = \int_{\Omega_0} J^{-1} \nabla N_i(\xi,\eta) \cdot J^{-1} \nabla N_j(\xi,\eta) |J| d\Omega_0$$

$$I = \int_{-1}^{1} \int_{-1}^{1} J^{-1} \nabla N_i(\xi,\eta) \cdot J^{-1} \nabla N_j(\xi,\eta) |J| d\xi \, d\eta$$

$$I \approx \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} w_{k_1} w_{k_2} J^{-1}_{k_1,k_2} \nabla N_i(\xi_{k_1}, \eta_{k_2}) \cdot J^{-1}_{k_1,k_2} \nabla N_j(\xi_{k_1}, \eta_{k_2}) |J_{k_1,k_2}|$$

$$M = \int_V N_i N_j dV$$

```
function I = Mass2DSymbolic()
    xi  = sym('xi', 'real');  eta = sym('eta', 'real');
    dx  = sym('dx', 'real');  dy  = sym('dy', 'real');

    c=[-1 -1; 1 -1; 1  1; -1  1];

    J(1,1) = 0.5*dx; J(2,2) = 0.5*dy;

    for i=1:4
        N(i) = 1/4*( 1+c(i,1)*xi )*( 1+c(i,2)*eta );
    end

    F   = det(J)*N*N';
    M   = int(int(F, 'xi', -1, 1), 'eta', -1, 1);
end
```

$$K = \int_V \nabla N_i \cdot \nabla N_j dV$$

```
function I = Laplace2DSymbolic()
    xi  = sym('xi', 'real'); eta = sym('eta', 'real');
    dx  = sym('dx', 'real'); dy  = sym('dy', 'real');
    u1 = sym('u1', 'real');  u2 = sym('u2', 'real');
    u3 = sym('u3', 'real');  u4 = sym('u4', 'real');

    c=[-1 -1; 1 -1; 1  1; -1  1];
    J(1,1) = 0.5*dx; J(2,2) = 0.5*dy;

    for i=1:4
        N(i) = 1/4*( 1+c(i,1)*xi )*( 1+c(i,2)*eta );
    end

    Nx  = diff(N, 'xi'); Ny  = diff(N, 'eta');
    dN = [Nx; Ny];
    U   = (u1*N(1) + u2*N(2) + u3*N(3) + u4*N(4))^2;
    Jd  = inv(J) * dN;
    F   = U*det(J)*Jd'*Jd;
    I   = int(int(F, 'xi', -1, 1), 'eta', -1, 1);
    I   = simplify(dx*dy*I);
end
```