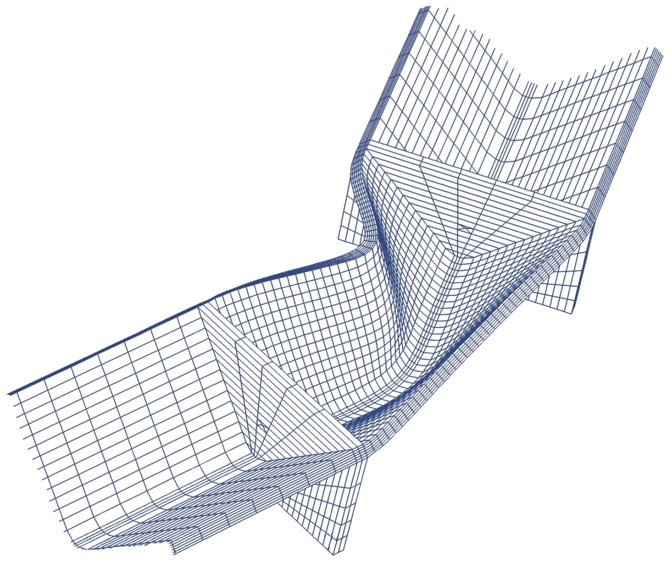
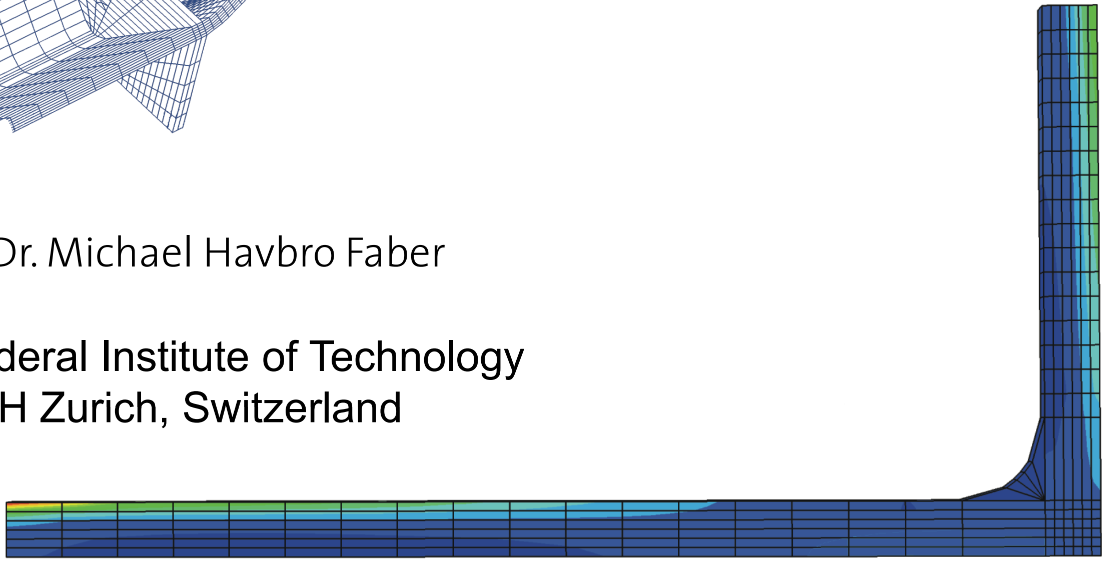
The Finite Element Method for the Analysis of Linear Systems



Prof. Dr. Michael Havbro Faber

Swiss Federal Institute of Technology
ETH Zurich, Switzerland



Contents of Today's Lecture

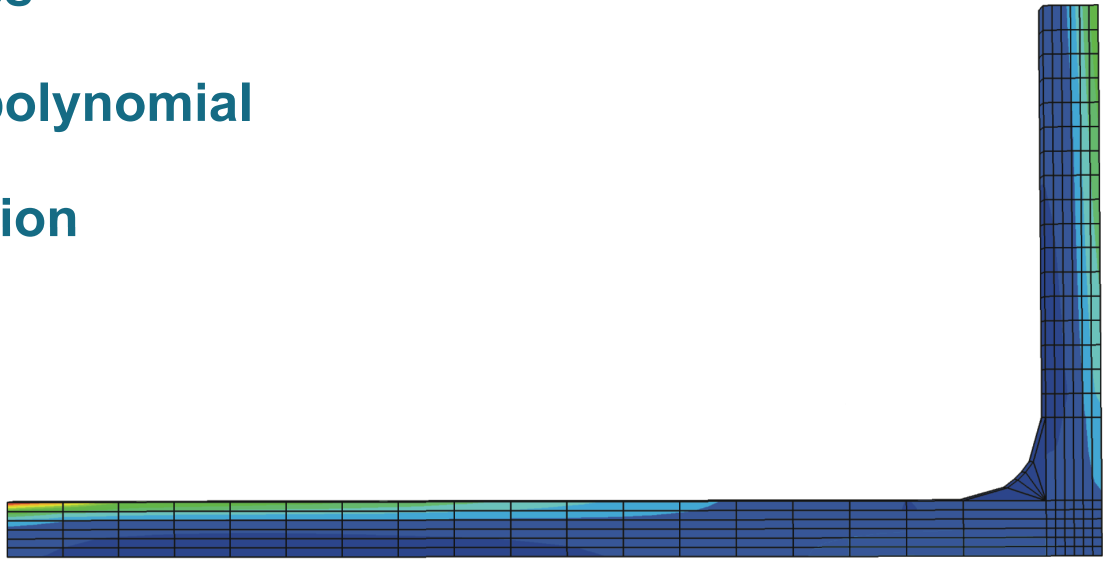
- Summary of last lecture
- The principle of iso-parametric finite elements
- Implementation of FEM

  - Integration of “matrixes”

  - Interpolation using a polynomial

  - Newton Cotes integration

  - Gauss integration



Summary of last lecture

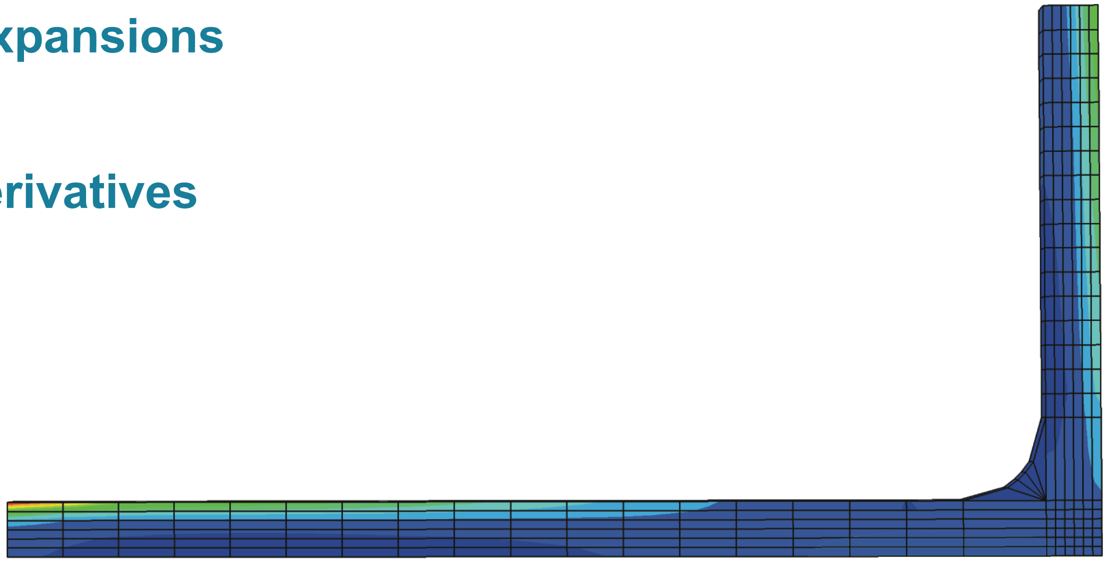
Shape functions:

Polynomials are usually applied for the development of shape functions (polynomials are easily differentiated analytically)

- Langrange polynomials
complete polynomial expansions

- Serendipity polynomials
incomplete polynomial expansions

- Hermitian polynomials
polynomials including derivatives



Summary of last lecture

Shape functions:

Lagrange polynomials (general):

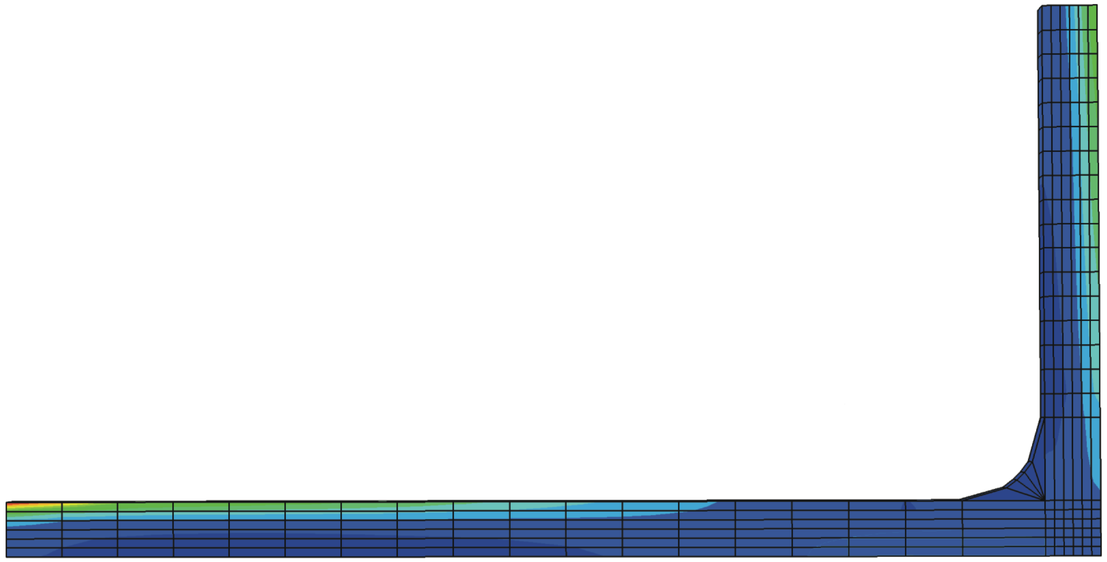
$$u(x,y,z)=\sum_{i=1}^{n} L_i(x,y,z)\hat{u}_i$$

$$u(x,y,z)=\sum_{i=1}^{n} \quad \hat{u}_i$$

$$L_i(x_j,y_j,z_j)=1, \quad i=j$$

$$L_i(x_j,y_j,z_j)=0, \quad i\neq j$$

$$\sum_{i=1}^{n} L_i(x,y,z)=1$$

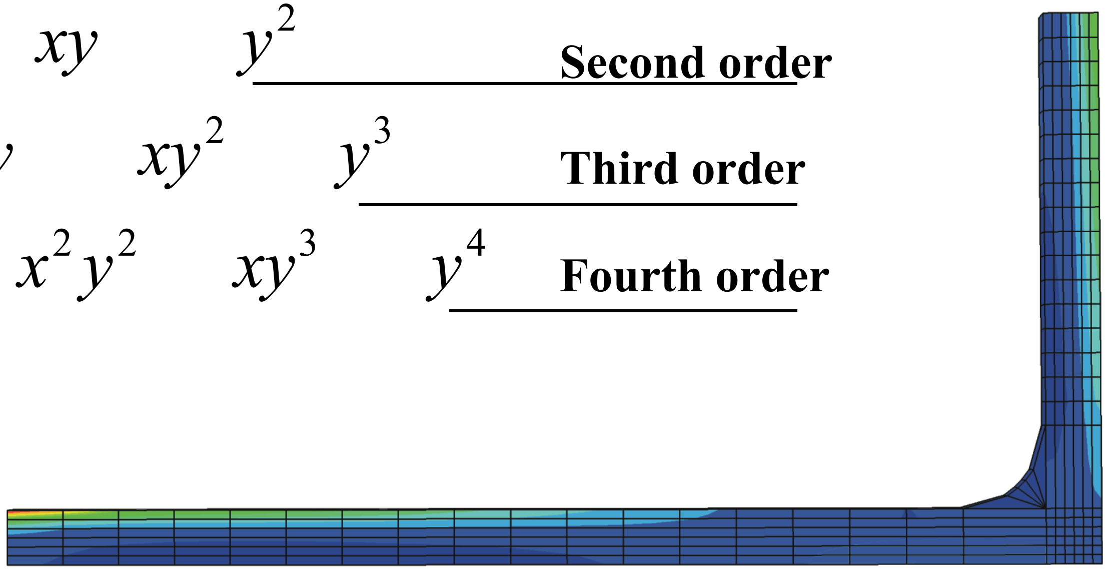


Summary of last lecture

Shape functions:

From Pascal's triangle we can see how many nodes are required for the representation of displacement fields of any order and completeness:

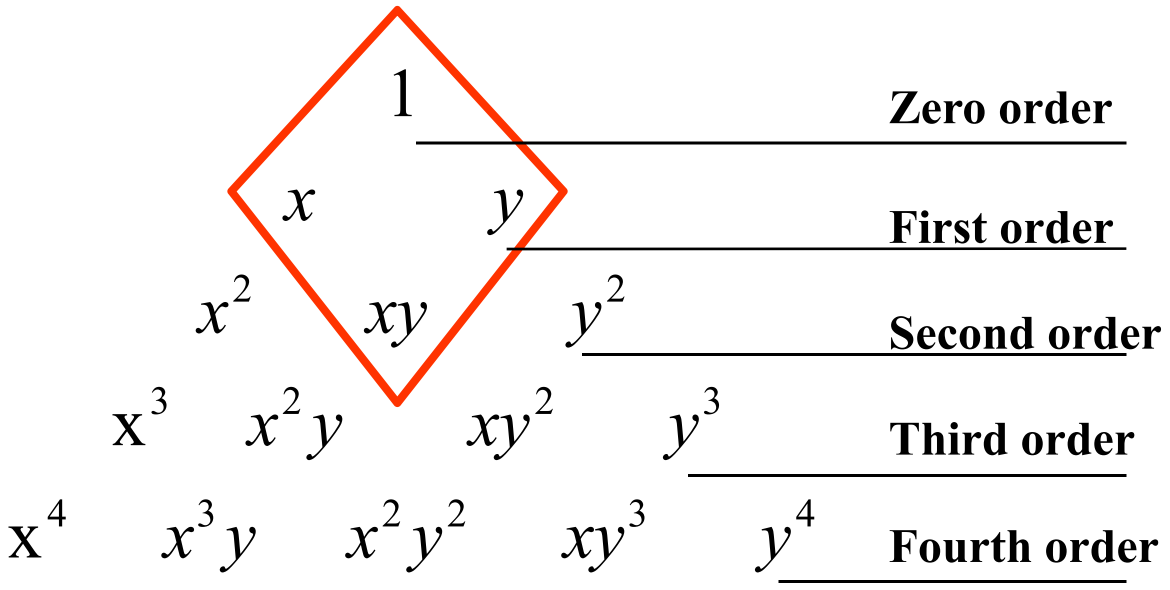
$$\begin{array}{ccccccccc|l}
 & & & & 1 & & & & & \textbf{Zero order} \\
 & & & x & & y & & & & \textbf{First order} \\
 & & x^2 & & xy & & y^2 & & & \textbf{Second order} \\
 & \mathrm{x}^3 & & x^2y & & xy^2 & & y^3 & & \textbf{Third order} \\
\mathrm{x}^4 & & x^3y & & x^2y^2 & & xy^3 & & y^4 & \textbf{Fourth order}
\end{array}$$

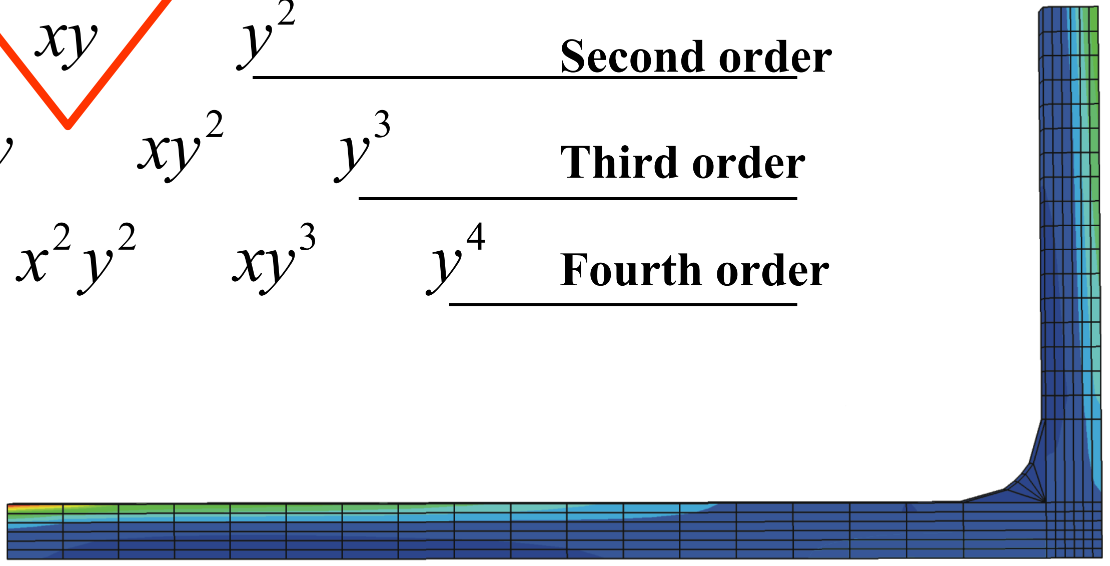


Summary of last lecture

Shape functions:

Products of Lagrange polynomials (bi-linear four node rectangular)



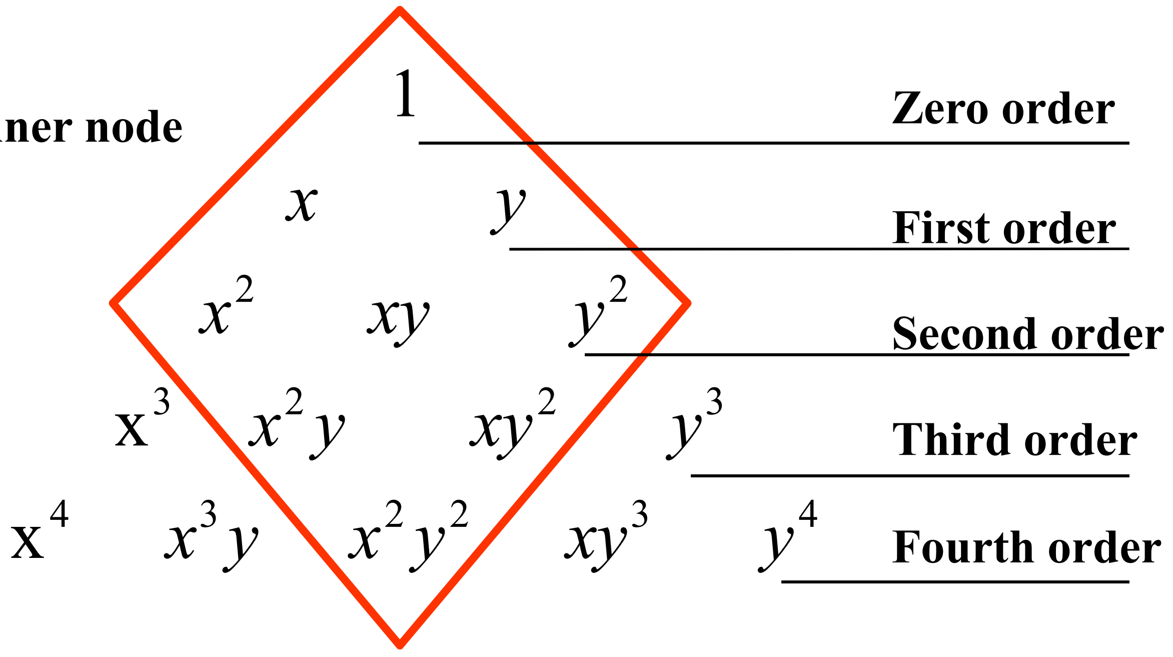


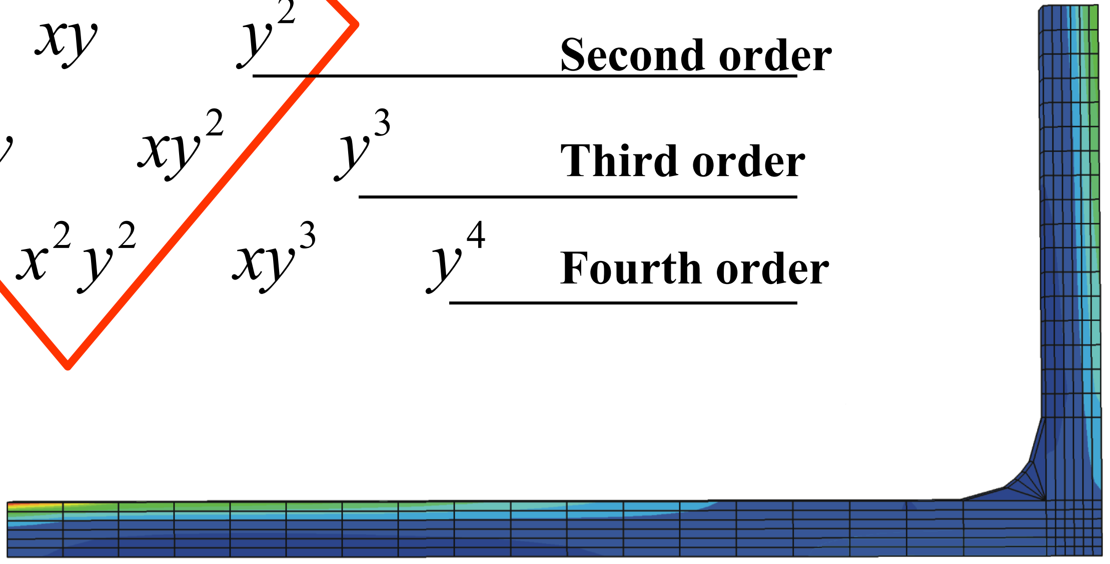
Summary of last lecture

Shape functions:

Products of Lagrange polynomials (quadratic nine-node rectangular)

This requires an inner node a difficulty !

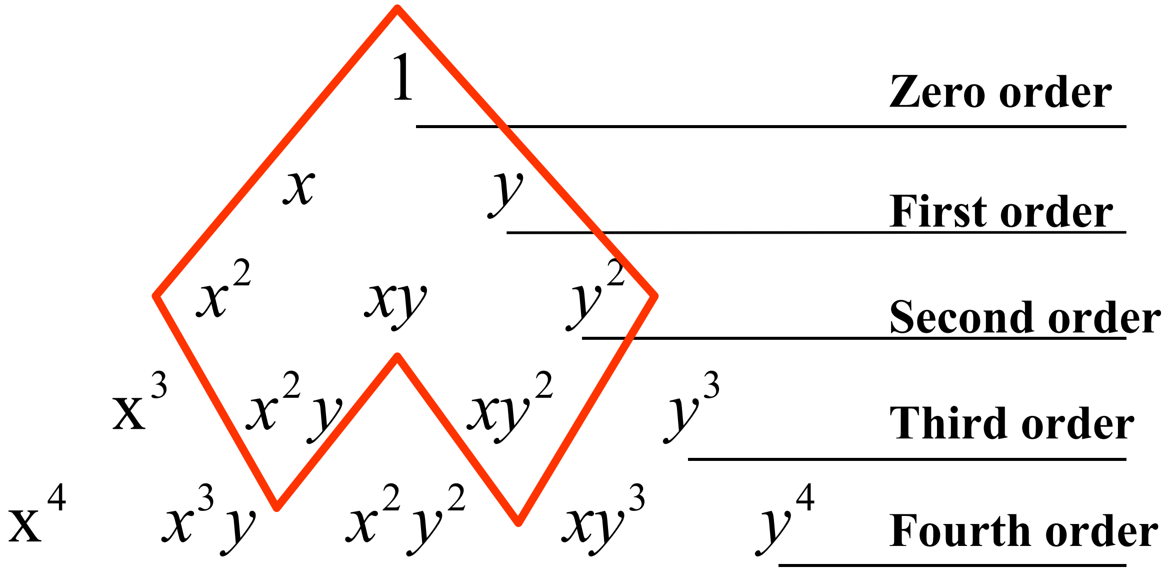


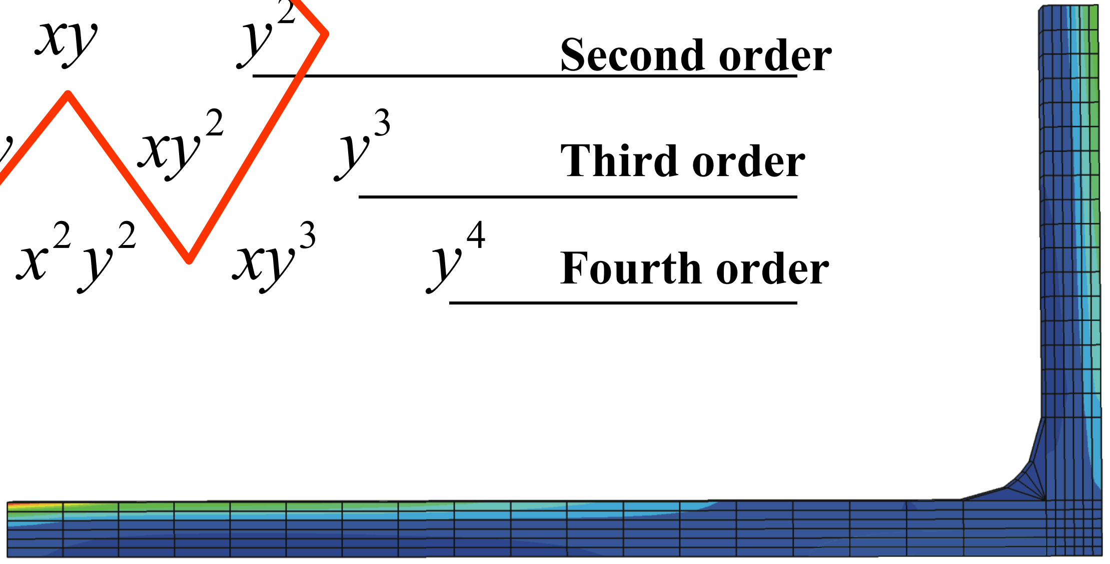


Summary of last lecture

Shape functions:

Serendipity shape functions are constructed by incomplete polynomials – avoiding inner nodes



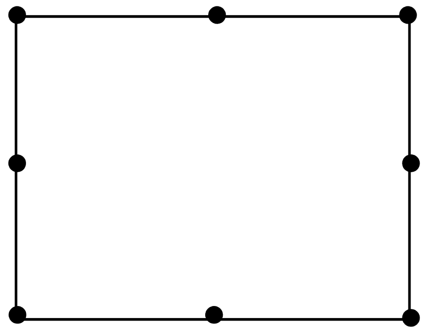


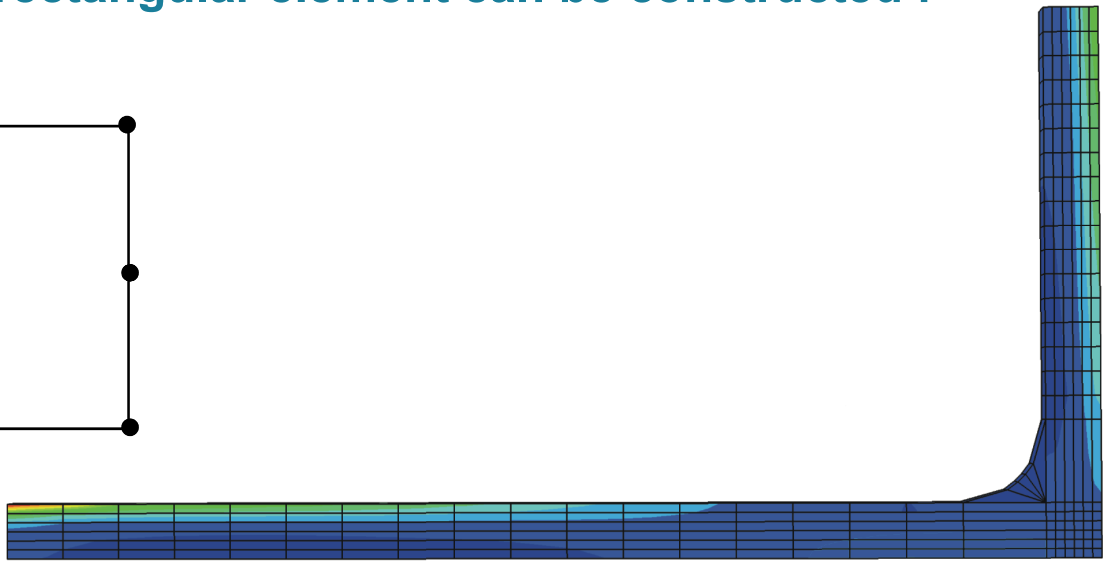
Summary of last lecture

Shape functions:

Whereas difficulties may arise (inner nodes) when aiming to develop quadratic shape functions for rectangular elements using Lagrange polynomials the shape functions developed by incomplete polynomials (**serendipity shape functions**) – less terms necessitates less nodes !

A bi-quadratic eight node rectangular element can be constructed !





Summary of last lecture

Shape functions:

Hermitian shape functions relate not only the displacements at nodes to displacements within the elements but also the first order derivatives

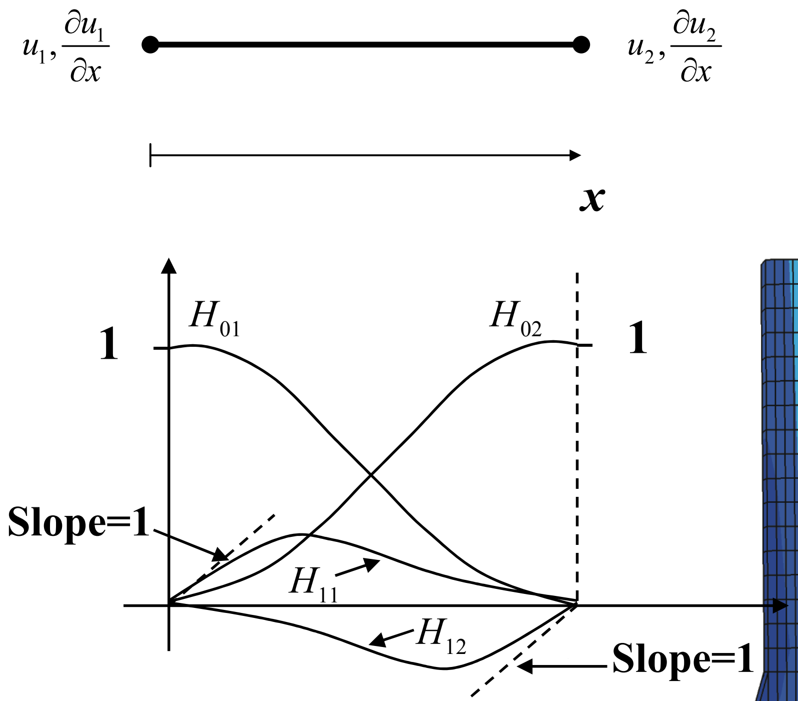
$$u(x)=\sum_{i=1}^{2}\left(H_{0i}(x)\hat{u}_i+H_{1i}(x)\frac{\partial\hat{u}_i}{\partial x}\right)$$

$H_{0i}(x)=1,$ and zero at the other node

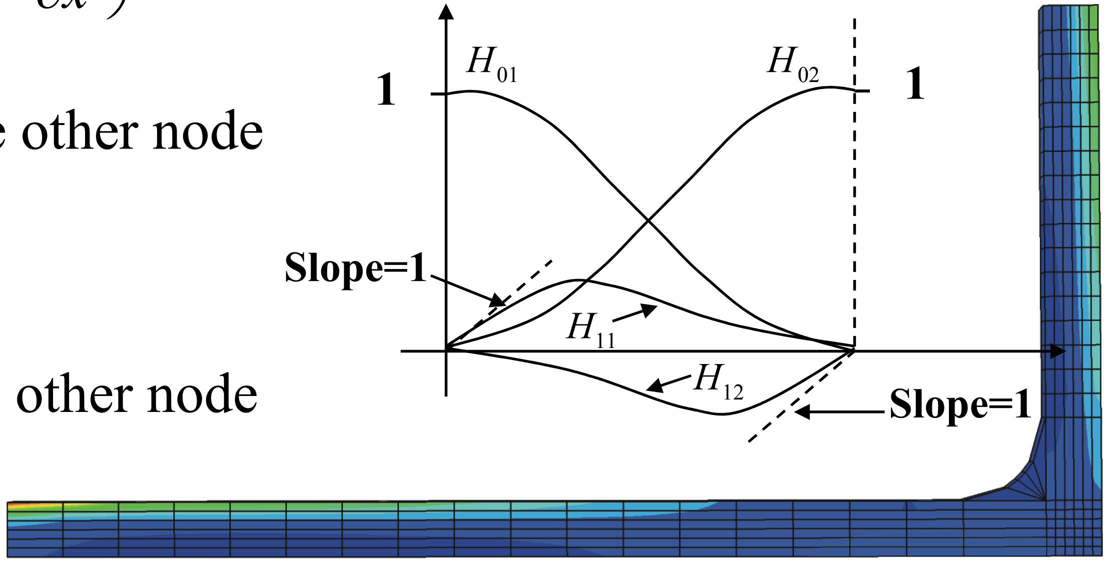
$H'_{0i}(x)=0$ at both nodes

$H_{1i}(x)=0,$ at both nodes

$H'_{1i}(x)=1,$ and zero at the other node



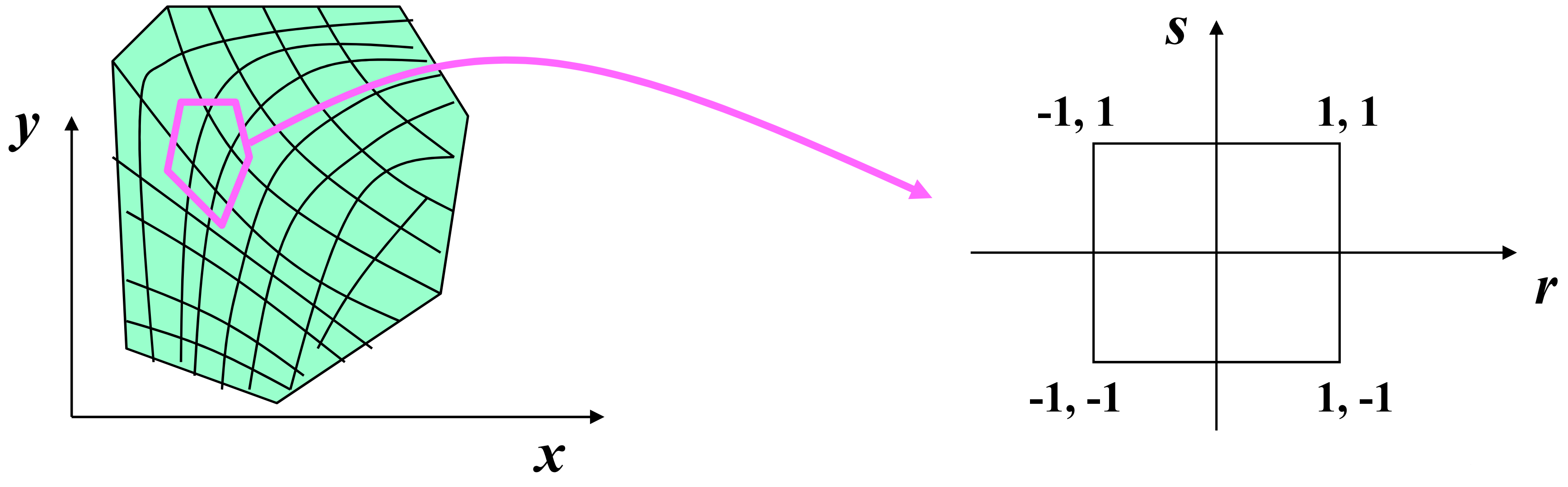
$u_1, \frac{\partial u_1}{\partial x}$
$u_2, \frac{\partial u_2}{\partial x}$
x
1
1
H_{01}
H_{02}
Slope=1
H_{11}
H_{12}
Slope=1

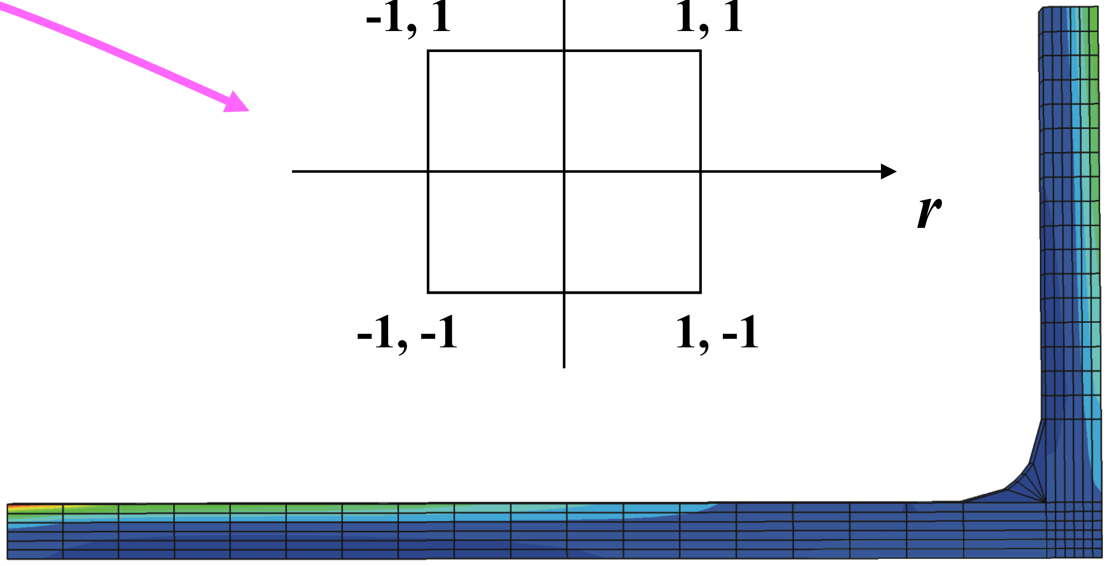


Summary of last lecture

Shape functions – Natural coordinates:

As we have seen we are able to establish shape functions in global or local coordinate systems as we please. However, for the purpose of standardizing the process of developing the element matrixes it is convenient to introduce the so-called natural coordinate system.

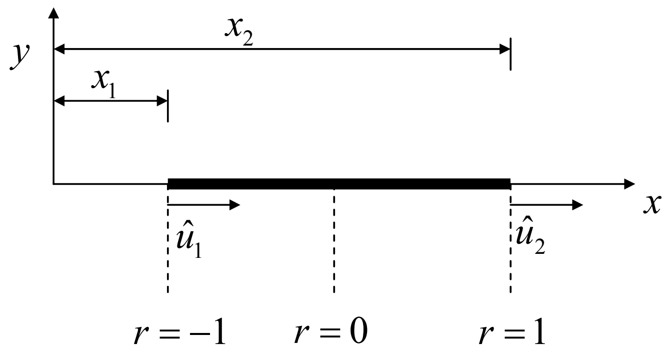




Summary of last lecture

Shape functions – Natural coordinates:

Let us consider the simple bar element



The relation between the x-coordinate and the r-coordinate is given as:

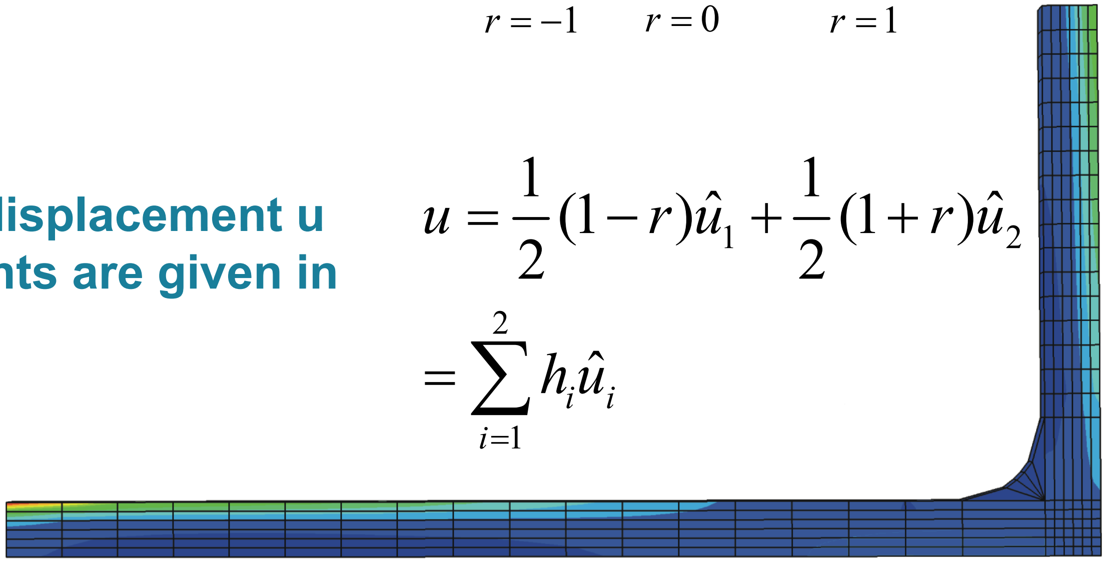
$$x = \frac{1}{2}(1-r)\hat{x}_1 + \frac{1}{2}(1+r)\hat{x}_2$$

$$= \sum_{i=1}^{2} h_i \hat{x}_i$$

The relation between the displacement u and the nodal displacements are given in the same way:

$$u = \frac{1}{2}(1-r)\hat{u}_1 + \frac{1}{2}(1+r)\hat{u}_2$$

$$= \sum_{i=1}^{2} h_i \hat{u}_i$$



The principle of iso-parametric FE

Shape functions – Natural coordinates:

Let us consider the simple bar element

We need to be able to establish the strains – meaning we need to be able to take the derivatives of the displacement filed in regard to the x-coordinate

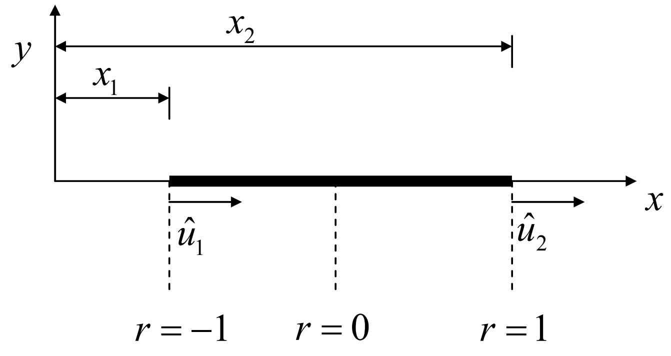
$$\varepsilon = \frac{du}{dx} = \frac{du}{dr}\frac{dr}{dx}$$

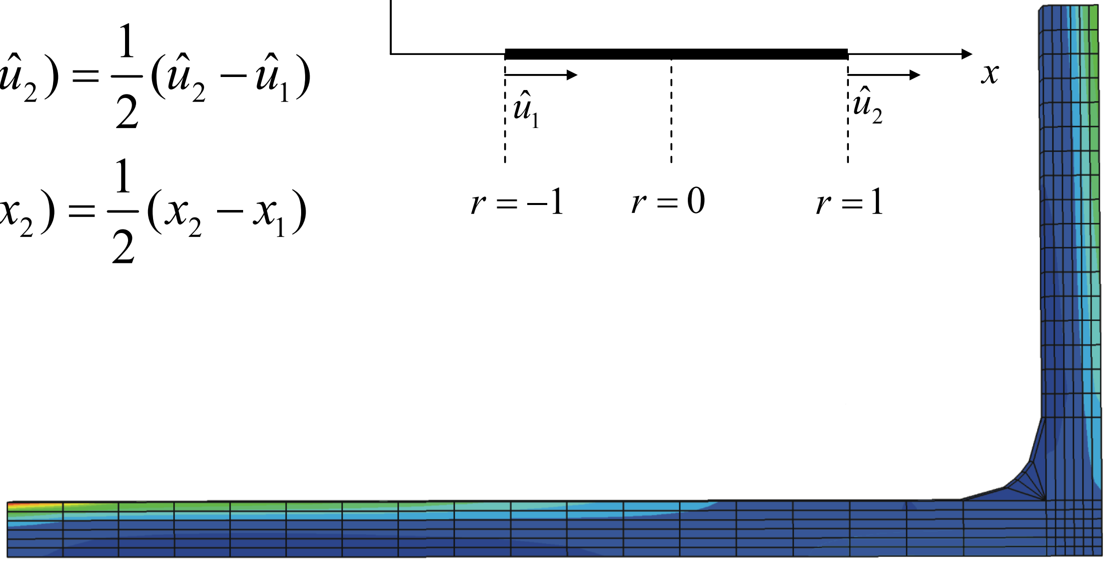
$$\frac{du}{dr} = \frac{d}{dr}\left(\frac{1}{2}(1-r)\hat{u}_1 + \frac{1}{2}(1+r)\hat{u}_2\right) = \frac{1}{2}(\hat{u}_2 - \hat{u}_1)$$

$$\frac{dx}{dr} = \frac{d}{dr}\left(\frac{1}{2}(1-r)x_1 + \frac{1}{2}(1+r)x_2\right) = \frac{1}{2}(x_2 - x_1)$$

$$\Downarrow$$

$$\frac{du}{dx} = \frac{(\hat{u}_2 - \hat{u}_1)}{(x_2 - x_1)} = \frac{(\hat{u}_2 - \hat{u}_1)}{L}$$





The principle of iso-parametric FE

Shape functions – Natural coordinates:

Let us consider the simple bar element

The strain-displacement matrix then becomes:

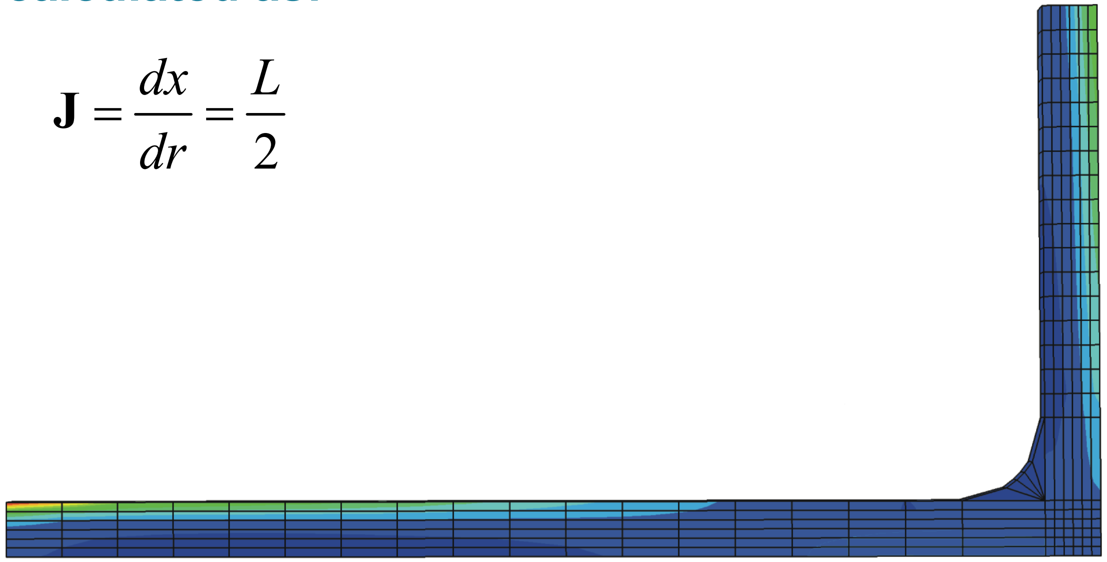
$$\mathbf{B} = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix}$$

and the stiffness matrix is calculated as:

$$\mathbf{K} = \frac{AE}{L^2}\int_{-1}^{1}\begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix}\mathbf{J}dr, \qquad \mathbf{J} = \frac{dx}{dr} = \frac{L}{2}$$

$$\Downarrow$$

$$\mathbf{K} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$



Implementation of FE

Finite Element Equilibrium Equations:

The equations we need to solve include several integrals !

$$\mathbf{K}\mathbf{U} = \mathbf{R}$$

$$\mathbf{K} = \sum_{m=1}^{N} \int_{V^{(m)}} \mathbf{B}^{(m)T}\mathbf{C}^{(m)}\mathbf{B}^{(m)} dV^{(m)}$$

$$\mathbf{R} = \mathbf{R}_B + \mathbf{R}_S - \mathbf{R}_I + \mathbf{R}_C$$

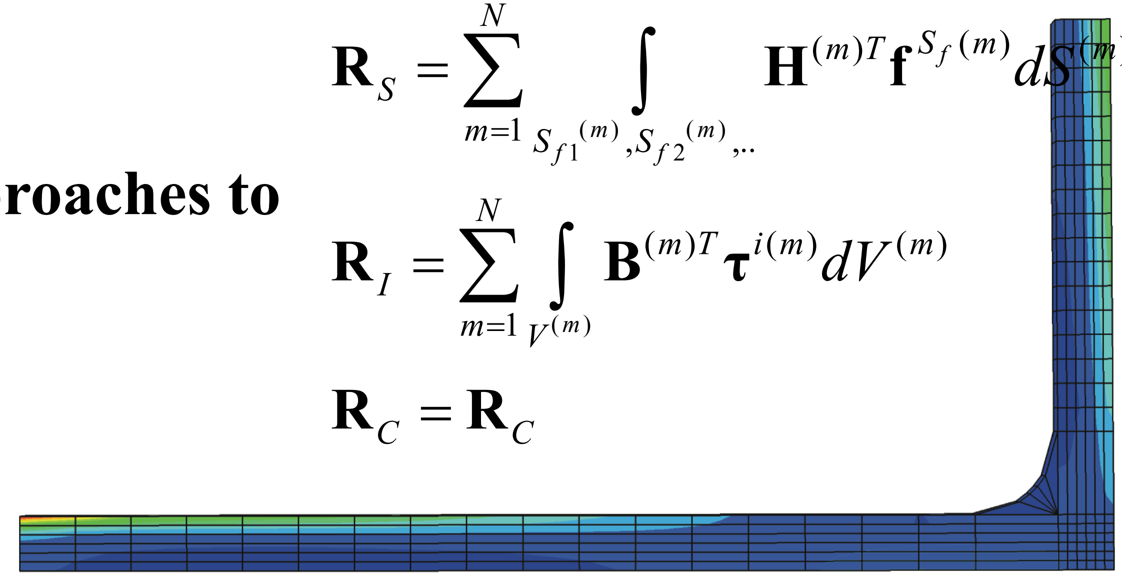
$$\mathbf{R}_B = \sum_{m=1}^{N} \int_{V^{(m)}} \mathbf{H}^{(m)T}\mathbf{f}^{B(m)} dV^{(m)}$$

$$\mathbf{R}_S = \sum_{m=1}^{N} \int_{S_{f1}^{(m)}, S_{f2}^{(m)}, ..} \mathbf{H}^{(m)T}\mathbf{f}^{S_f(m)} dS^{(m)}$$

$$\mathbf{R}_I = \sum_{m=1}^{N} \int_{V^{(m)}} \mathbf{B}^{(m)T}\boldsymbol{\tau}^{i(m)} dV^{(m)}$$

$$\mathbf{R}_C = \mathbf{R}_C$$

We need efficient approaches to solve these integrals



Implementation of FE

In principle we need to consider 1, 2 and 3 dimensional integrals

We may write the integrals in the following way

$$\int \mathbf{F}(r)dr, \quad \int \mathbf{F}(r,s)drds, \quad \int \mathbf{F}(r,s,t)drdsdt$$

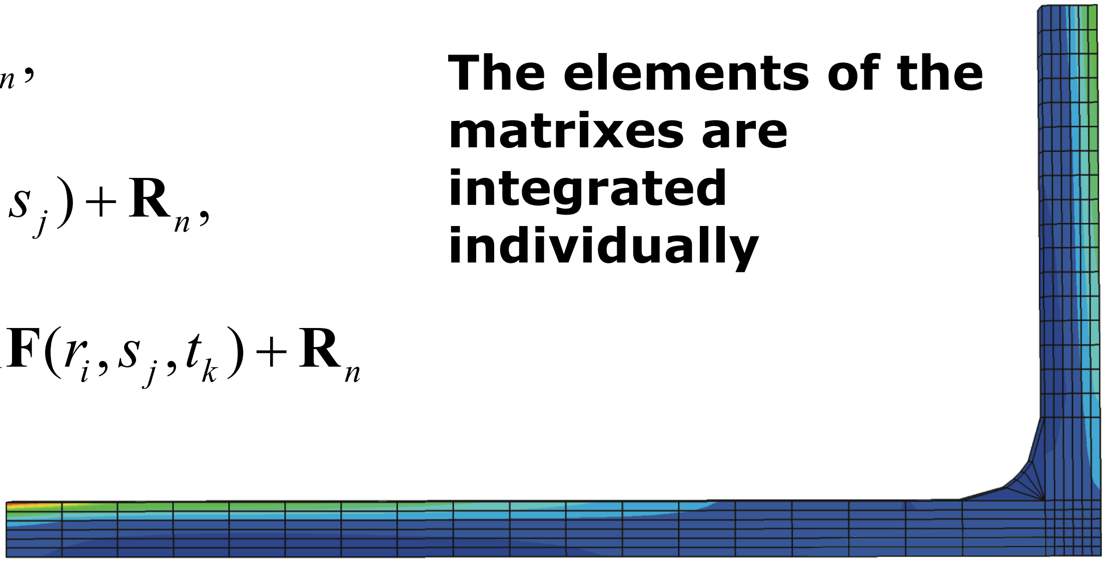
In practice we may solve the integrals in terms of sums

$$\int \mathbf{F}(r)dr = \sum_i \alpha_i \mathbf{F}(r_i) + \mathbf{R}_n,$$

$$\int \mathbf{F}(r,s)drds = \sum_{i,j} \alpha_{ij} \mathbf{F}(r_i, s_j) + \mathbf{R}_n,$$

$$\int \mathbf{F}(r,s,t)drdsdt = \sum_{i,j,k} \alpha_{ijk} \mathbf{F}(r_i, s_j, t_k) + \mathbf{R}_n$$

The elements of the matrixes are integrated individually



Implementation of FE

In principle we need to consider 1, 2 and 3 dimensional integrals

We may write the integrals in the following way

$$\int \mathbf{F}(r)dr, \quad \int \mathbf{F}(r,s)drds, \quad \int \mathbf{F}(r,s,t)drdsdt$$

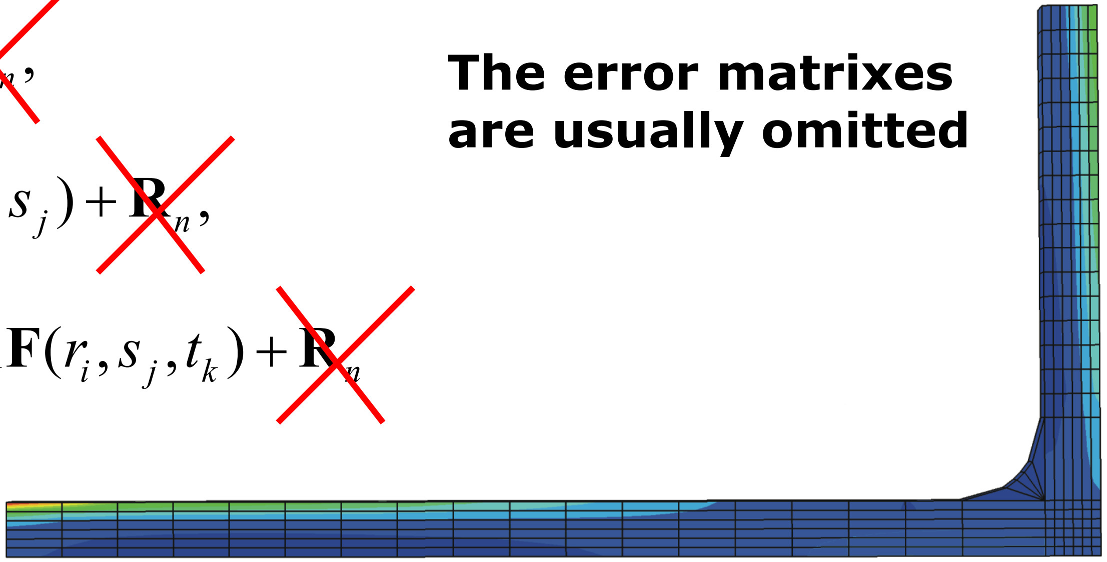
In practice we may solve the integrals in terms of sums

$$\int \mathbf{F}(r)dr = \sum_i \alpha_i \mathbf{F}(r_i) + \cancel{\mathbf{R}_n},$$

$$\int \mathbf{F}(r,s)drds = \sum_{i,j} \alpha_{ij} \mathbf{F}(r_i, s_j) + \cancel{\mathbf{R}_n},$$

$$\int \mathbf{F}(r,s,t)drdsdt = \sum_{i,j,k} \alpha_{ijk} \mathbf{F}(r_i, s_j, t_k) + \cancel{\mathbf{R}_n}$$

The error matrixes are usually omitted



Implementation of FE

In principle we need to consider 1, 2 and 3 dimensional integrals

We may write the integrals in the following way

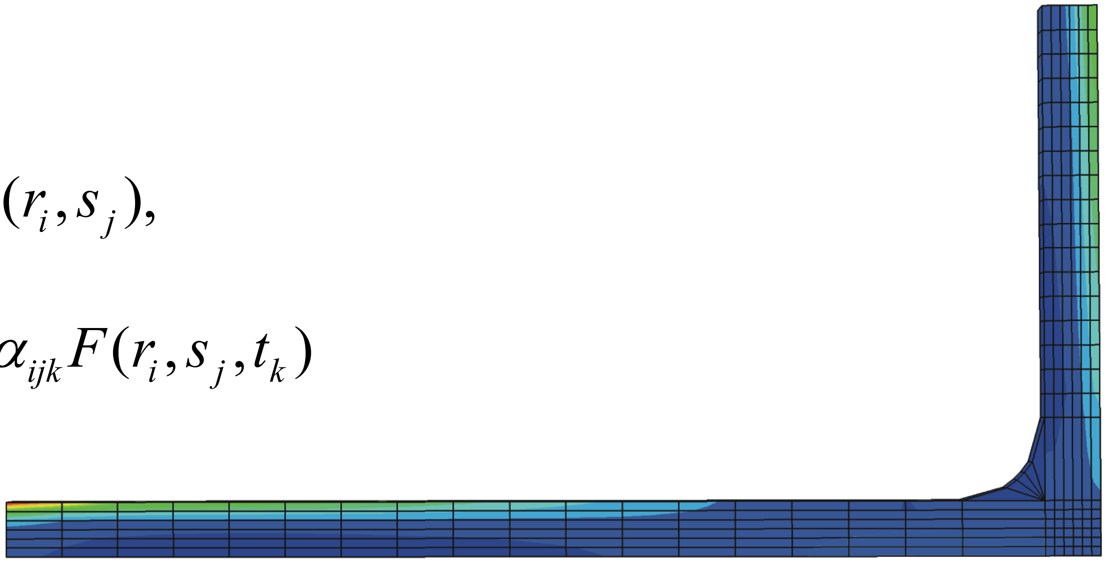
$$\int \mathbf{F}(r)dr, \quad \int \mathbf{F}(r,s)drds, \quad \int \mathbf{F}(r,s,t)drdsdt$$

The elements of the matrixes are integrated individually

$$\int F(r)dr = \sum_{i} \alpha_i F(r_i),$$

$$\int F(r,s)drds = \sum_{i,j} \alpha_{ij} F(r_i, s_j),$$

$$\int F(r,s,t)drdsdt = \sum_{i,j,k} \alpha_{ijk} F(r_i, s_j, t_k)$$



Implementation of FE

In principle we need to consider 1, 2 and 3 dimensional integrals

Considering the 1-dimensional case

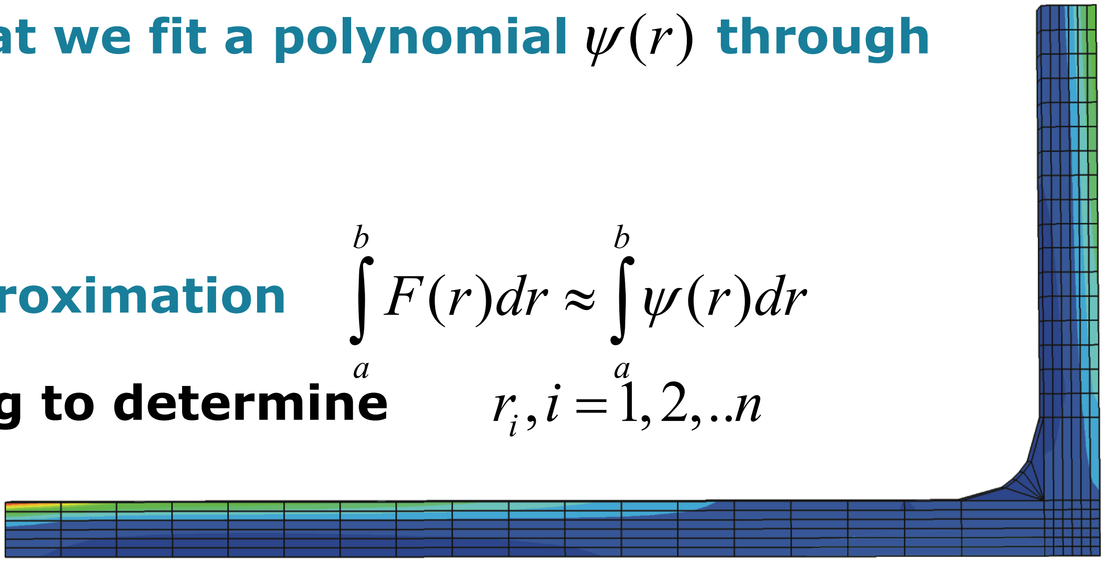
$$\int_a^b F(r)dr, \quad \text{iso-parametric; } a=-1, b=1$$

The general idea is that we fit a polynomial $\psi(r)$ through

$$F(r_i), i=1,2,..n$$

and introduce the approximation $\int_a^b F(r)dr \approx \int_a^b \psi(r)dr$

The problem remaining to determine $r_i, i=1,2,..n$



Implementation of FE

Interpolation using a polynomial

Having calculated $F(r_i), \quad r_i, i = 0,1,2,..n$

we may fit a unique polynomial through these values

$$\psi(r) = a_0 + a_1 r + a_2 r^2 + \ldots + a_n r^n$$

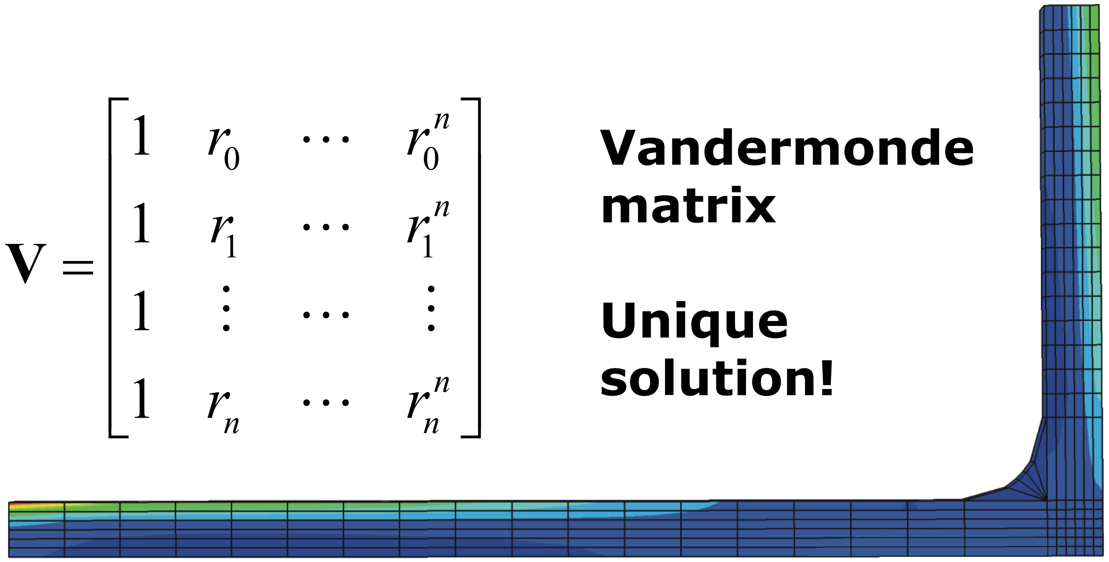
we get

$$\mathbf{F} = \mathbf{V}\mathbf{a},$$

$$\mathbf{F} = \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_n \end{bmatrix}, \qquad \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} 1 & r_0 & \cdots & r_0^n \\ 1 & r_1 & \cdots & r_1^n \\ 1 & \vdots & \cdots & \vdots \\ 1 & r_n & \cdots & r_n^n \end{bmatrix}$$

Vandermonde matrix

Unique solution!



Implementation of FE

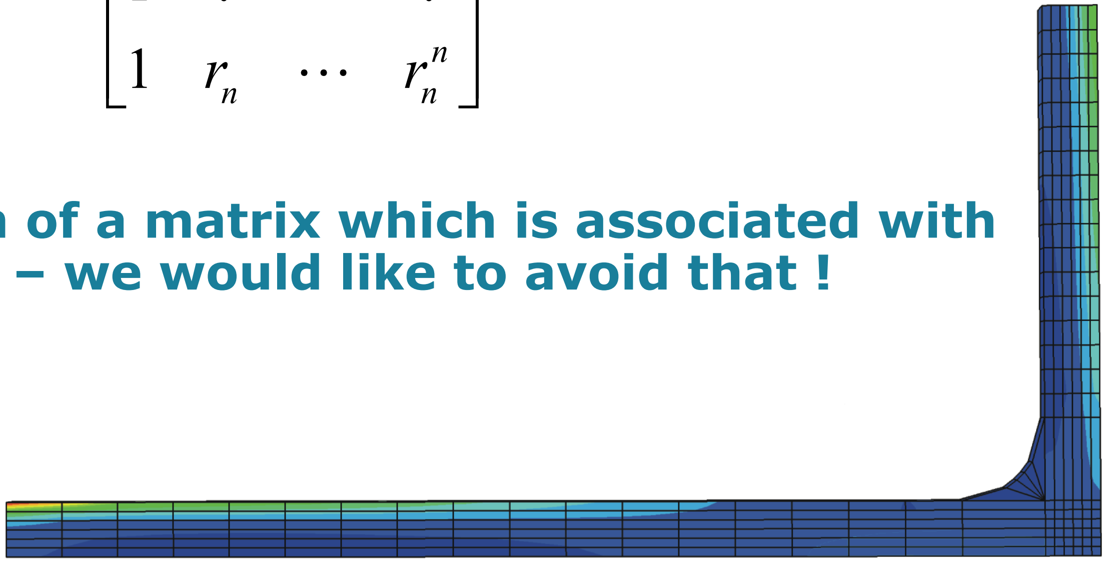
Interpolation using a polynomial

To solve

$$\mathbf{F} = \mathbf{V}\mathbf{a},$$

$$\mathbf{F} = \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_n \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} 1 & r_0 & \cdots & r_0^n \\ 1 & r_1 & \cdots & r_1^n \\ 1 & \vdots & \cdots & \vdots \\ 1 & r_n & \cdots & r_n^n \end{bmatrix}$$

Requires the inversion of a matrix which is associated with some numerical effort – we would like to avoid that !



Implementation of FE

Interpolation using a polynomial

We may associate with the functions

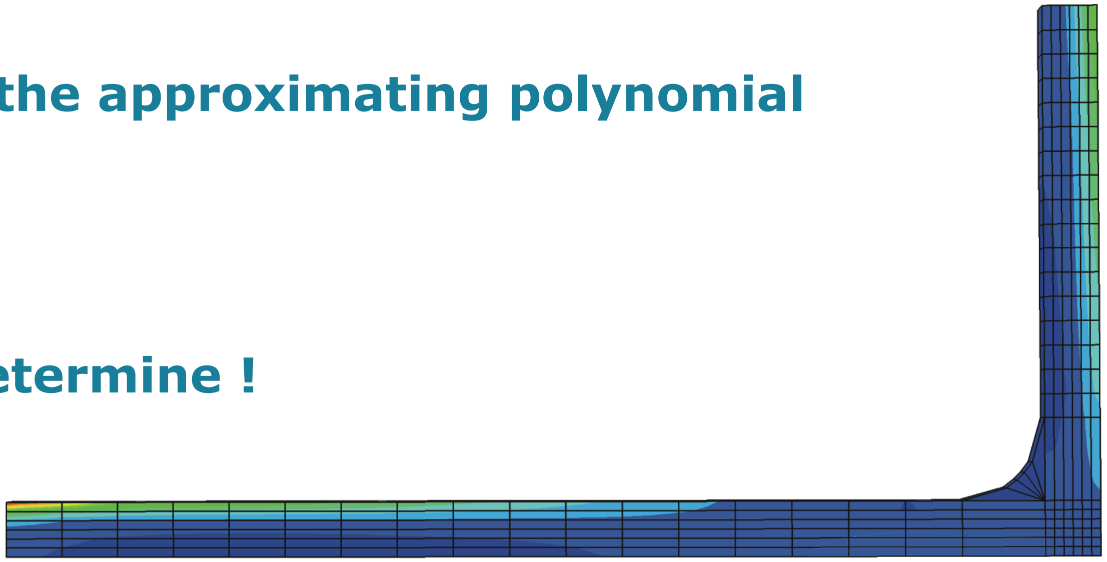
$$1, r, r^2, \ldots, r^n$$

the axes of a *n*+1 dimensional vector space in which the specific coordinates

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

define the approximating polynomial

however difficult to determine !



Implementation of FE

Langrangian interpolation functions

Instead of using the basis $1, r, r^2, \ldots, r^n$

we may use Lagrangian interpolation functions of the form

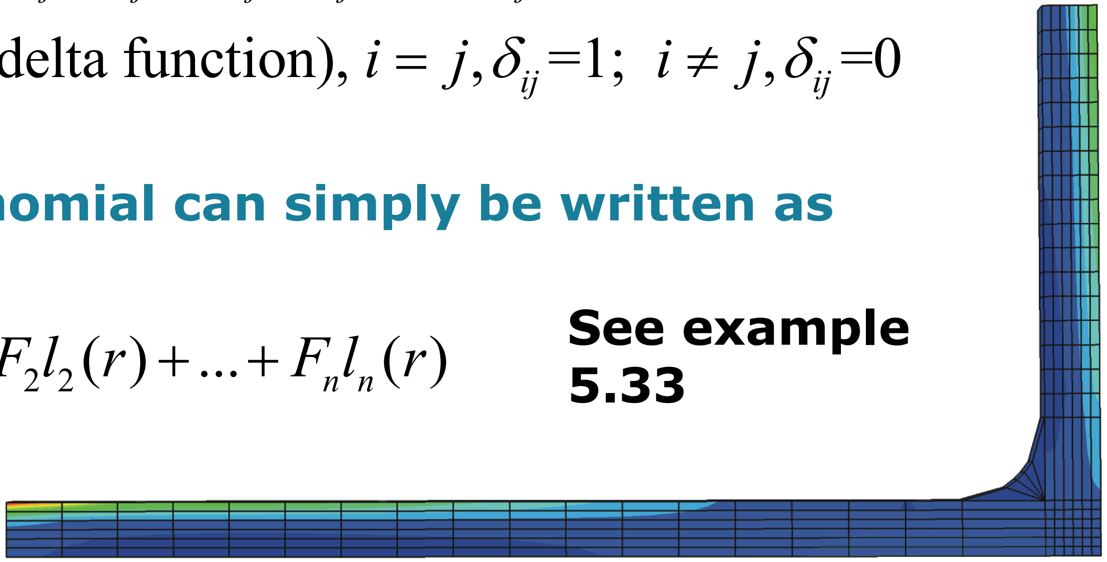
$$l_j(r) = \frac{(r-r_0)(r-r_1)\cdots(r-r_{j-1})(r-r_{j+1})\cdots(r-r_n)}{(r_j-r_0)(r_j-r_1)\cdots(r_j-r_{j-1})(r_j-r_{j+1})\cdots(r_j-r_n)}$$

$$l_j(r_i) = \delta_{ij} \quad \text{(Kroneckers delta function)}, \; i = j, \delta_{ij}=1; \;\; i \neq j, \delta_{ij}=0$$

in this space the polynomial can simply be written as

$$\psi(r) = F_0 l_0(r) + F_1 l_1(r) + F_2 l_2(r) + \ldots + F_n l_n(r)$$

See example 5.33



Implementation of FE

Newton-Cotes integration

We assume equidistantly space integration points, i.e. for a one dimensional integral between *a* and *b* we get

$$r_0 = a; \quad r_n = b; \quad h = \frac{b-a}{n}$$

$$\int_a^b F(r)dr = \sum_{i=0}^{n}\left[\int_a^b l_i(r)dr\right]F_i + R_n$$

$$= (b-a)\sum_{i=0}^{n}\left[C_i^n\right]F_i + R_n$$

$\uparrow$

The Newton-Cotes constants

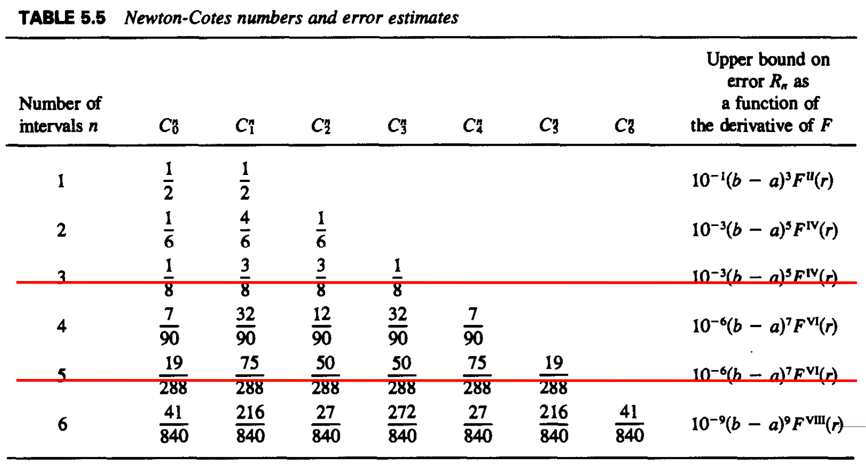
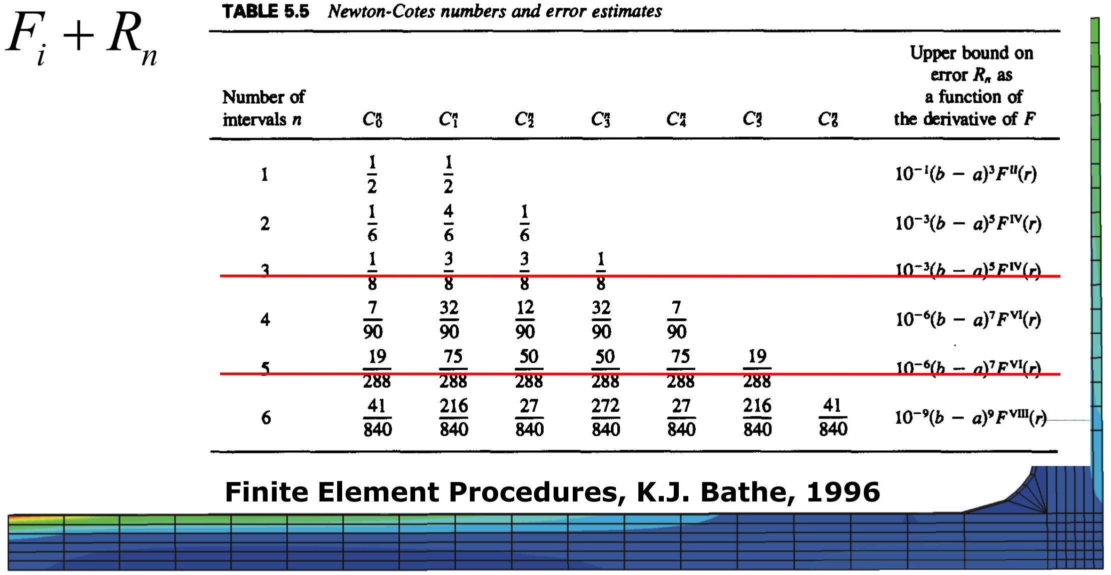


TABLE 5.5 *Newton-Cotes numbers and error estimates*

Number of intervals n	C_0^n	C_1^n	C_2^n	C_3^n	C_4^n	C_5^n	C_6^n	Upper bound on error R_n as a function of the derivative of F
1	$\frac{1}{2}$	$\frac{1}{2}$						$10^{-1}(b-a)^3F^{II}(r)$
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$					$10^{-3}(b-a)^5F^{IV}(r)$
~~3~~	~~$\frac{1}{8}$~~	~~$\frac{3}{8}$~~	~~$\frac{3}{8}$~~	~~$\frac{1}{8}$~~				~~$10^{-3}(b-a)^5F^{IV}(r)$~~
4	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$			$10^{-6}(b-a)^7F^{VI}(r)$
~~5~~	~~$\frac{19}{288}$~~	~~$\frac{75}{288}$~~	~~$\frac{50}{288}$~~	~~$\frac{50}{288}$~~	~~$\frac{75}{288}$~~	~~$\frac{19}{288}$~~		~~$10^{-6}(b-a)^7F^{VI}(r)$~~
6	$\frac{41}{840}$	$\frac{216}{840}$	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{216}{840}$	$\frac{41}{840}$	$10^{-9}(b-a)^9F^{VIII}(r)$

Finite Element Procedures, K.J. Bathe, 1996



Implementation of FE

Newton-Cotes integration

Using the Newton-Cotes approach we may increase the precision by

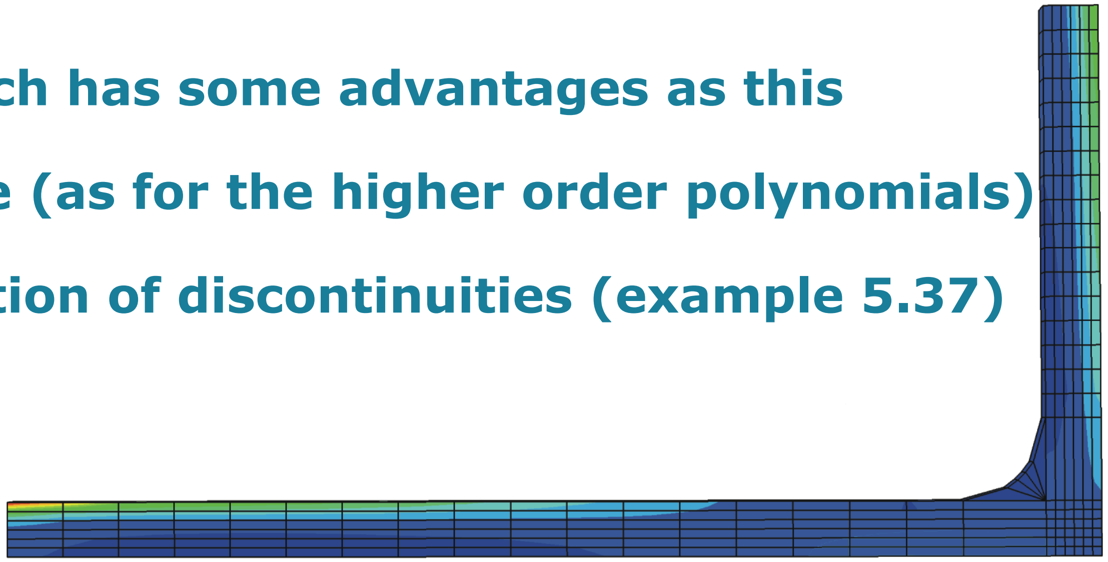
- increasing the number of intervals – higher order polynomial

- subdividing the integral into parts (composite approach)

the composite approach has some advantages as this

- ensures convergence (as for the higher order polynomials)

- allows for consideration of discontinuities (example 5.37)



Implementation of FE

Gauss integration

In the foregoing we assumed equidistant integration points

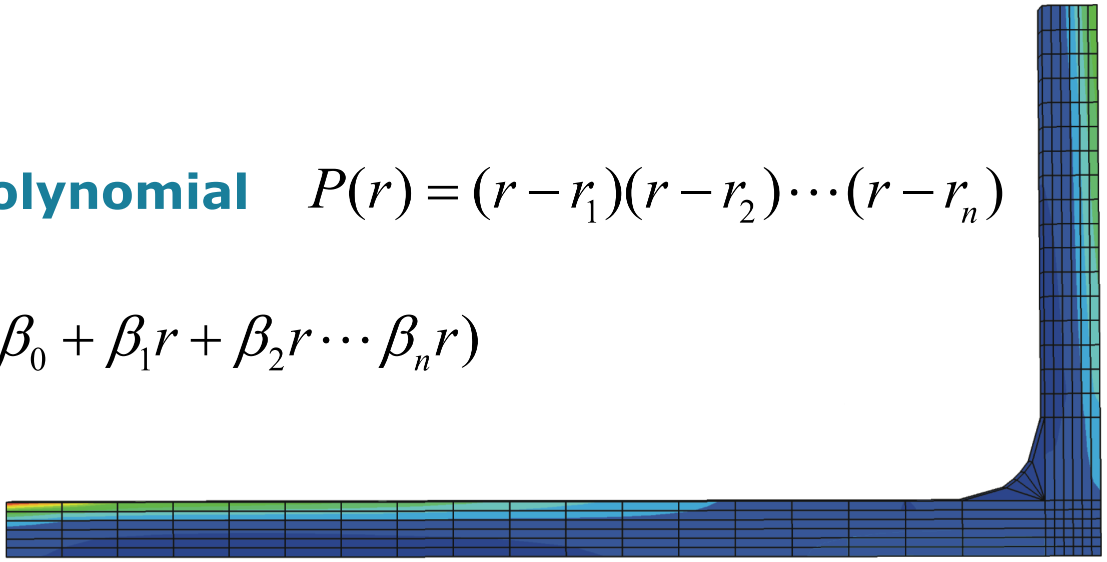
In the following we will not only optimize the number of integration points but also the location (distance between) the integration points

As before we can write

$$\psi(r) = \sum_{j=1}^{n} F_j l_j(r)$$

but now we add the polynomial $P(r) = (r - r_1)(r - r_2)\cdots(r - r_n)$

$$F(r) = \sum_{j=1}^{n} F_j l_j(r) + P(r)(\beta_0 + \beta_1 r + \beta_2 r \cdots \beta_n r)$$



Implementation of FE

Gauss integration

$$F(r) = \sum_{j=1}^{n} F_j l_j(r) + P(r)(\beta_0 + \beta_1 r + \beta_2 r \cdots \beta_n r)$$

By integrating we get

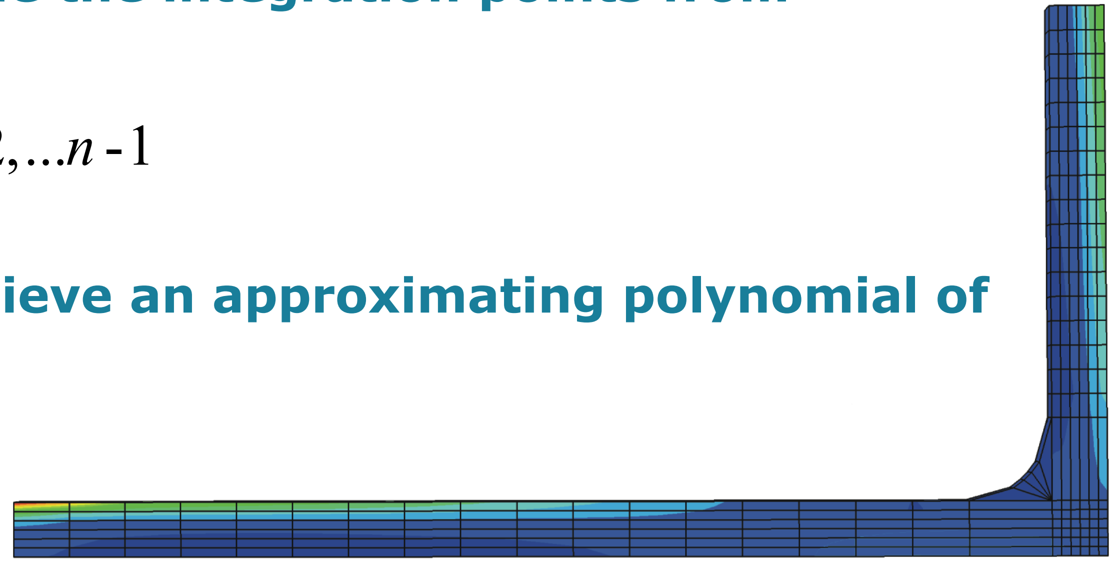
$$\int_a^b F(r)dr = \sum_{i=1}^{n}\left[\int_a^b l_i(r)dr\right]F_i + \sum_{j=0}^{n-1}\beta_j\left[\int_a^b r^j P(r)\right]$$

we may now determine the integration points from

$$\int_a^b P(r)r^k dr = 0, \quad k = 0,1,2,\ldots n-1$$

and in the end we achieve an approximating polynomial of order

$$2n-1$$



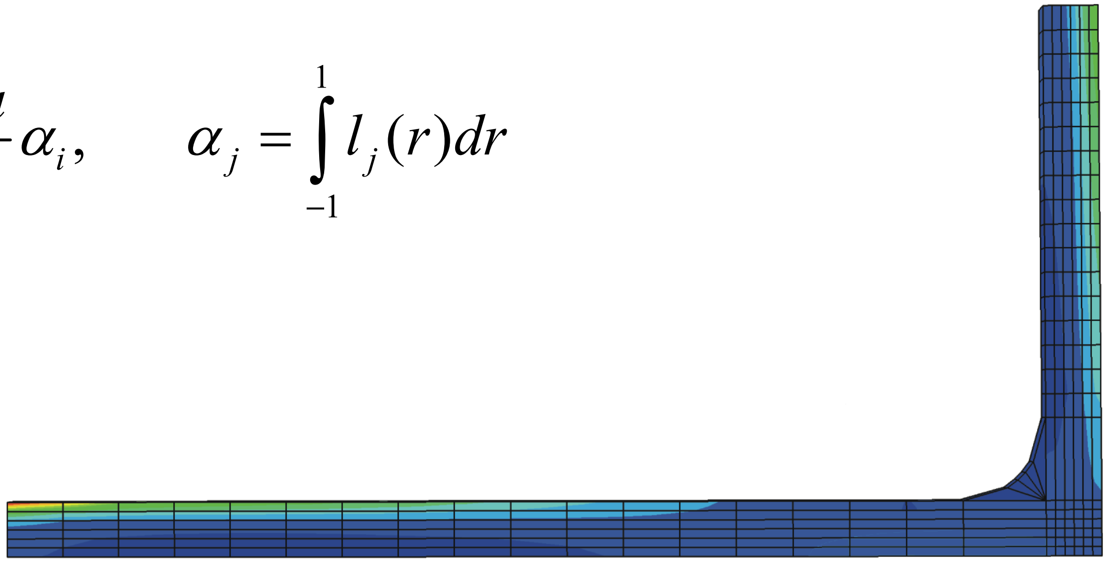
Implementation of FE

Gauss integration

The sampling weights clearly depend on the integration limits and for this reason it is obviously beneficial to standardize the integration domain.

This may easily be achieved by integrating from -1 to +1 and then to adjust the sampling points and weights as:

$$\frac{a+b}{2}+\frac{b-a}{2}r_i, \qquad \frac{b-a}{2}\alpha_i, \qquad \alpha_j=\int_{-1}^{1} l_j(r)dr$$



Implementation of FE

Gauss integration

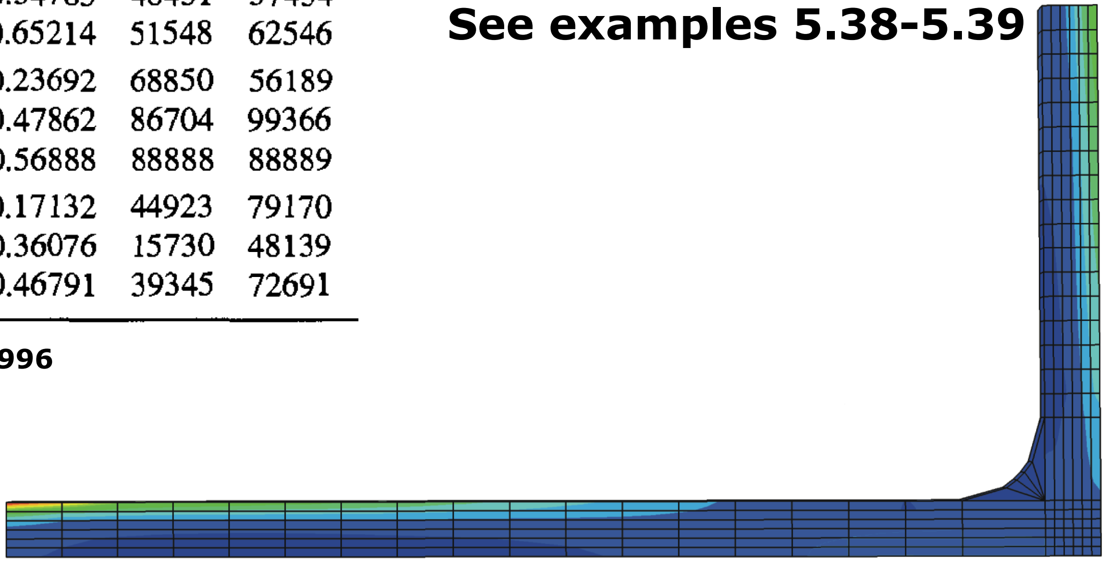
$$\frac{a+b}{2}+\frac{b-a}{2}r_i, \qquad \frac{b-a}{2}\alpha_i, \qquad \alpha_j=\int_{-1}^{1} l_j(r)dr$$

TABLE 5.6 *Sampling points and weights in Gauss-Legendre numerical integration (interval −1 to +1)*

n	r_i			α_i		
1	0.	(15 zeros)		2.	(15 zeros)	
2	±0.57735	02691	89626	1.00000	00000	00000
3	±0.77459	66692	41483	0.55555	55555	55556
	0.00000	00000	00000	0.88888	88888	88889
4	±0.86113	63115	94053	0.34785	48451	37454
	±0.33998	10435	84856	0.65214	51548	62546
5	±0.90617	98459	38664	0.23692	68850	56189
	±0.53846	93101	05683	0.47862	86704	99366
	0.00000	00000	00000	0.56888	88888	88889
6	±0.93246	95142	03152	0.17132	44923	79170
	±0.66120	93864	66265	0.36076	15730	48139
	±0.23861	91860	83197	0.46791	39345	72691

Finite Element Procedures, K.J. Bathe, 1996

See examples 5.38-5.39



Implementation of FE

Gauss integration

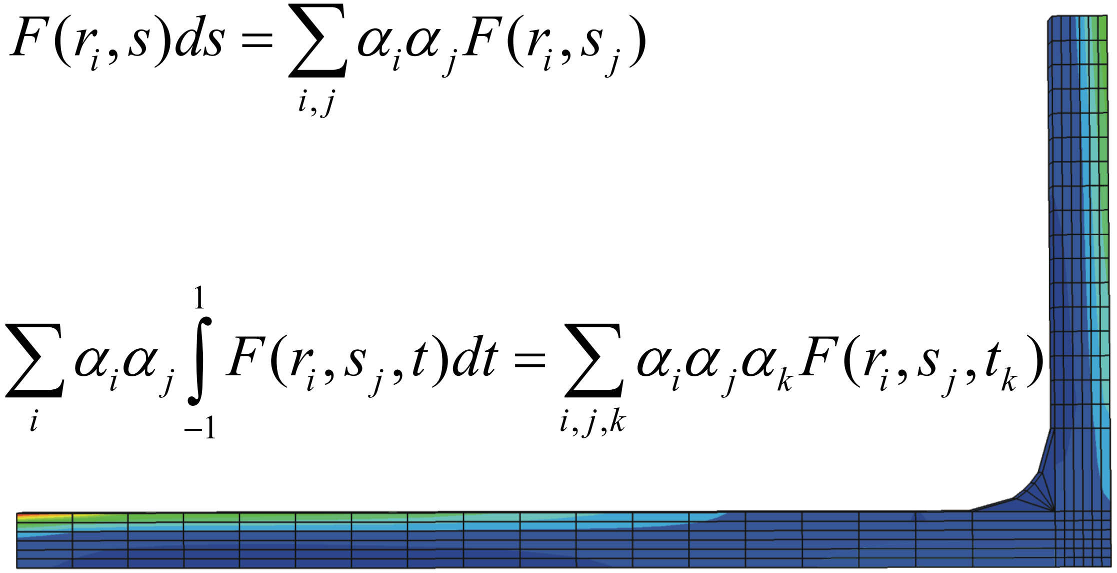
So far we looked at 1-dimensional integrals – but the same principle applies to 2 and 3-dimensional integrals as well

2-dimensions

$$\int_{-1}^{1}\int_{-1}^{1} F(r,s)drds = \sum_i \alpha_i \int_{-1}^{1} F(r_i,s)ds = \sum_{i,j} \alpha_i \alpha_j F(r_i,s_j)$$

3-dimensions

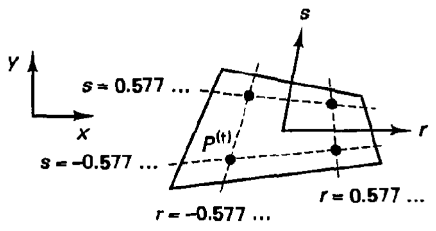
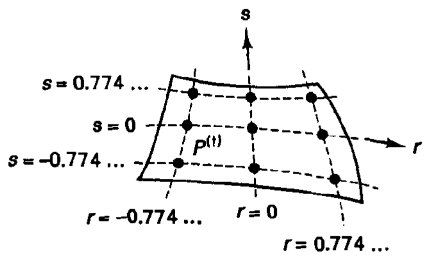
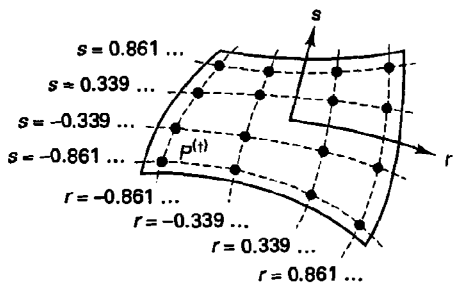
$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} F(r,s,t)drdsdt = \sum_i \alpha_i \alpha_j \int_{-1}^{1} F(r_i,s_j,t)dt = \sum_{i,j,k} \alpha_i \alpha_j \alpha_k F(r_i,s_j,t_k)$$



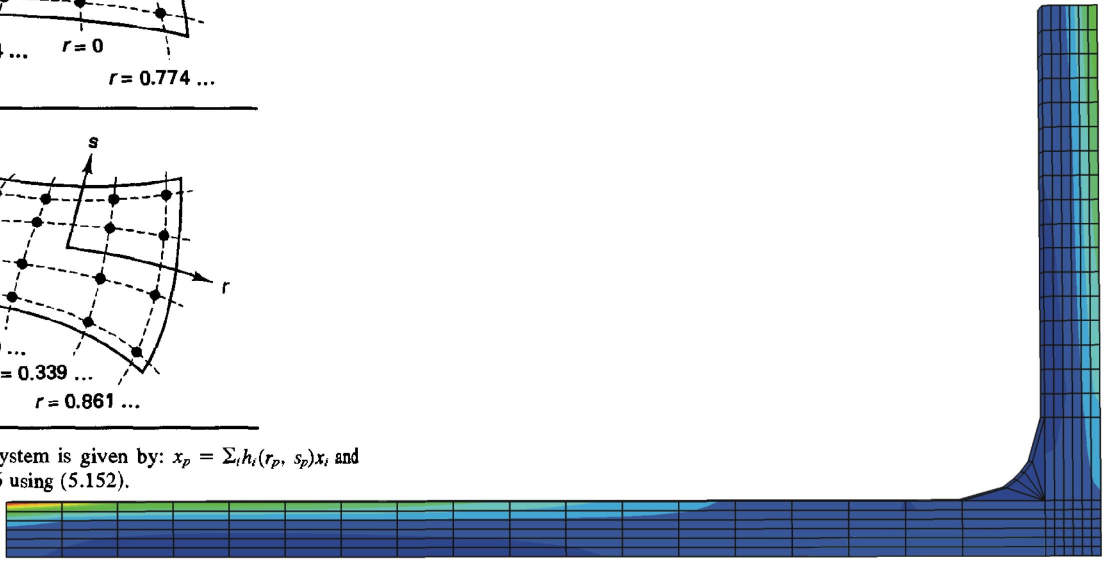
Implementation of FE

Finite Element Procedures, K.J. Bathe, 1996

TABLE 5.7 *Gauss numerical integrations over quadrilateral domains*

Integration order	Degree of precision	Location of integration points
2 × 2	3	
3 × 3	5	
4 × 4	7	

(†) The location of any integration point in the x, y coordinate system is given by: $x_p = \Sigma_i h_i(r_p, s_p)x_i$ and $y_p = \Sigma_i h_i(r_p, s_p)y_i$. The integration weights are given in Table 5.6 using (5.152).

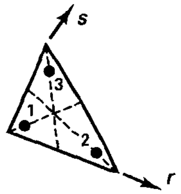
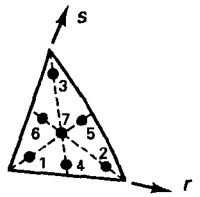
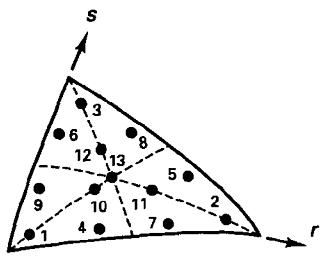


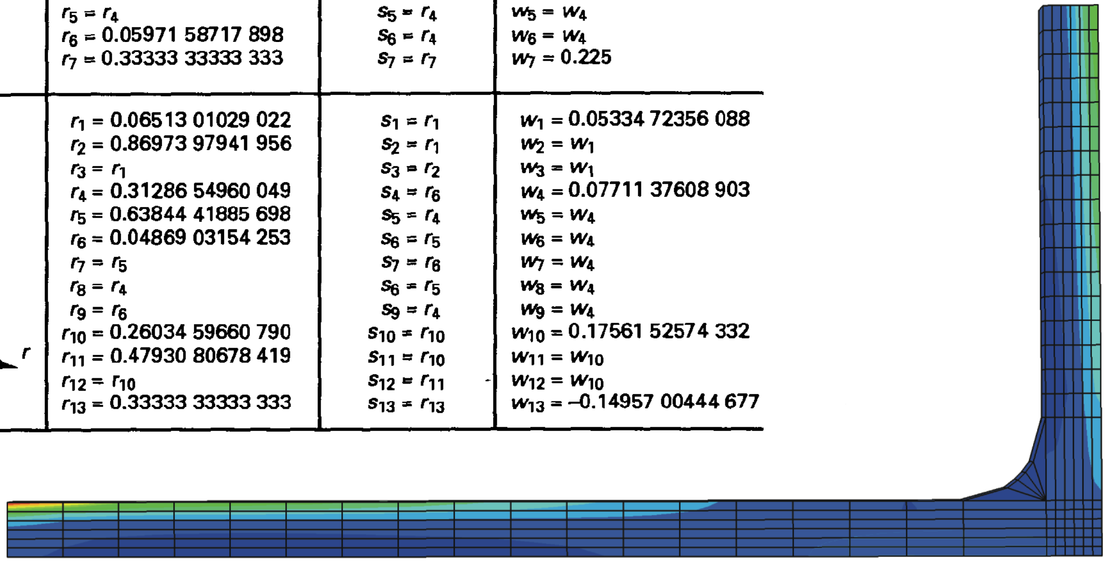
Implementation of FE

Gauss integration

Finite Element Procedures, K.J. Bathe, 1996

TABLE 5.8 *Gauss numerical integrations over triangular domains* $[\iint F\,dr\,ds = \frac{1}{2}\sum w_i F(r_i, s_i)]$

Integration order	Degree of precision	Integration points	r-coordinates	s-coordinates	Weights
3-point	2		$r_1 = 0.16666\ 66666\ 667$ $r_2 = 0.66666\ 66666\ 667$ $r_3 = r_1$	$s_1 = r_1$ $s_2 = r_1$ $s_3 = r_2$	$w_1 = 0.33333\ 33333\ 333$ $w_2 = w_1$ $w_3 = w_1$
7-point	5		$r_1 = 0.10128\ 65073\ 235$ $r_2 = 0.79742\ 69853\ 531$ $r_3 = r_1$ $r_4 = 0.47014\ 20641\ 051$ $r_5 = r_4$ $r_6 = 0.05971\ 58717\ 898$ $r_7 = 0.33333\ 33333\ 333$	$s_1 = r_1$ $s_2 = r_1$ $s_3 = r_2$ $s_4 = r_6$ $s_5 = r_4$ $s_6 = r_4$ $s_7 = r_7$	$w_1 = 0.12593\ 91805\ 448$ $w_2 = w_1$ $w_3 = w_1$ $w_4 = 0.13239\ 41527\ 885$ $w_5 = w_4$ $w_6 = w_4$ $w_7 = 0.225$
13-point	7	 s 3 6 8 12 13 5 9 10 11 2 1 4 7 r	$r_1 = 0.06513\ 01029\ 022$ $r_2 = 0.86973\ 97941\ 956$ $r_3 = r_1$ $r_4 = 0.31286\ 54960\ 049$ $r_5 = 0.63844\ 41885\ 698$ $r_6 = 0.04869\ 03154\ 253$ $r_7 = r_5$ $r_8 = r_4$ $r_9 = r_6$ $r_{10} = 0.26034\ 59660\ 790$ $r_{11} = 0.47930\ 80678\ 419$ $r_{12} = r_{10}$ $r_{13} = 0.33333\ 33333\ 333$	$s_1 = r_1$ $s_2 = r_1$ $s_3 = r_2$ $s_4 = r_6$ $s_5 = r_4$ $s_6 = r_5$ $s_7 = r_6$ $s_6 = r_5$ $s_9 = r_4$ $s_{10} = r_{10}$ $s_{11} = r_{10}$ $s_{12} = r_{11}$ $s_{13} = r_{13}$	$w_1 = 0.05334\ 72356\ 088$ $w_2 = w_1$ $w_3 = w_1$ $w_4 = 0.07711\ 37608\ 903$ $w_5 = w_4$ $w_6 = w_4$ $w_7 = w_4$ $w_8 = w_4$ $w_9 = w_4$ $w_{10} = 0.17561\ 52574\ 332$ $w_{11} = w_{10}$ $w_{12} = w_{10}$ $w_{13} = -0.14957\ 00444\ 677$



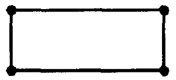
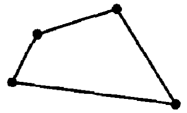
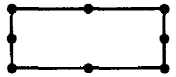
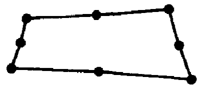
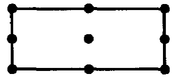
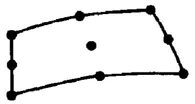
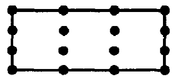
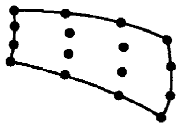
Finite Element Procedures, K.J. Bathe, 1996

Implementation of FE

Gauss integration

Integration order

TABLE 5.9 *Recommended full Gauss numerical integration orders for the evaluation of isoparametric displacement-based element matrices (use of Table 5.7)*

Two-dimensional elements (plane stress, plane strain and axisymmetric conditions)		Integration order
4-node		2 × 2
4-node distorted		2 × 2
8-node		3 × 3
8-node distorted		3 × 3
9-node		3 × 3
9-node distorted		3 × 3
16-node		4 × 4
16-node distorted		4 × 4

(Note: In axisymmetric analysis, the hoop strain effect is in all cases not integrated exactly, but with sufficient accuracy.)