Numerical Methods in Engineering

Interpolation (Shape Functions)
Interpolation Problem 1D

Let’s assume that we have measured an unknown magnitude $u = u(x)$ on a set of data points $x = [x_1, x_2, ..., x_N]$ obtaining a set of values $u = [u_1, u_2, ..., u_N]$.
Let’s assume that we have measured an unknown magnitude $u = u(x)$ on a set of data points $x = [x_1, x_2, \ldots, x_N]$ obtaining a set of values $u = [u_1, u_2, \ldots, u_N]$.

How can we guess the value of a new point?
Interpolation Problem 1D

Let’s assume that we have measured an unknown magnitude $u = u(x)$ on a set of data points $x = [x_1, x_2, \ldots, x_N]$ obtaining a set of values $u = [u_1, u_2, \ldots, u_N]$

 Actual function  
$f(x) = x^3 - 3x^2 - 3\sin(5x)$
Interpolation Problem 1D

To **Interpolate** a set of data measured points \((x_i, u_i)\) means to build a function \(P_n(x)\) (usually a **polynomial**) passing through these points. That is

\[
u_i = P_n(x_i), \ i = 1 \ldots N\]

**Theorem:**
The interpolation problem **has a unique** solution when \(n = N - 1\).

**Idea of the proof:**
\[P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n\]
\[
u_i = P_n(x_i), \ i = 1 \ldots N\]

**Unique** solution if the system is compatible determined. That means the interpolation polynomial has **degree one less** than the number of points.
Shape Functions 1D
Shape Functions 1D

Lagrange Polynomials (Shape functions)

One (N-1) degree Polynomial for each measure point \( x_i, \ i = 1, \ldots, N \)

\[
\psi_i(x) = \frac{(x - x_1)(x - x_2) \ldots (x - x_i) \ldots (x - x_{N-1})(x - x_N)}{(x_i - x_1)(x_i - x_2) \ldots (x_i - x_i) \ldots (x_i - x_{N-1})(x_i - x_N)}
\]

Properties:

1. Lagrange Polynomials are a base of the set \( \mathbb{P}_n[x] \)
2. \( \psi_i(x_j) = 0 \) if \( i \neq j \)
3. \( \psi_i(x_i) = 1 \)

Interpolation Polynomial (a linear combination of the shape functions)

\[
P_n(x) = u_1\psi_1(x) + u_2\psi_2(x) + \ldots + u_N\psi_N(x)
\]
Shape Functions 1D

N = 2 only two points: $x = [x_1, x_2]$ and their measures $u = [u_1, u_2]$

$$\psi_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}$$

$x = [1, 2]$

It is a straight line
Shape Functions 1D

$N = 2$ only two points: $x = [x_1, x_2]$ and their measures $u = [u_1, u_2]$

$$\psi_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}$$

$$\psi_2(x) = \frac{(x - x_1)}{(x_2 - x_1)}$$

$x = [1, 2]$
Shape Functions 1D

$N = 2$ only two points: $x = [x_1, x_2]$ and their measures $u = [u_1, u_2]$

$\psi_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}$

$\psi_2(x) = \frac{(x - x_1)}{(x_2 - x_1)}$

Interpolation Polynomial

$P_1(x) = u_1 \psi_1(x) + u_2 \psi_2(x)$

It is also a straight line (linear interpolation)
Shape Functions 1D

\( N = 2 \) only two points: \( x = [x_1, x_2] \) and their measures \( u = [u_1, u_2] \)

\[
\psi_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} \\
\psi_2(x) = \frac{(x - x_1)}{(x_2 - x_1)}
\]

Interpolation Polynomial

\[
P_1(x) = u_1 \psi_1(x) + u_2 \psi_2(x)
\]

Example:

In the case \( x = [1, 2] \) and \( u = [0.7, 1.8] \)

\[
\psi_1(x) = \frac{(x - 2)}{(1 - 2)} = -(x - 2) \\
\psi_2(x) = \frac{(x - 1)}{(2 - 1)} = (x - 1)
\]

How to approximate the value for \( x = 1.5 \)?

\[
P_1(1.5) = 0.7\psi_1(1.5) + 1.8\psi_2(1.5) \\
= 0.7 \times 0.5 + 1.8 \times 0.5 = 1.25
\]
Shape Functions 1D

Three points (N=3): \( x = [x_1, x_2, x_3] \) and measures \( u = [u_1, u_2, u_3] \)

\[
\psi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \\
\psi_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} \\
\psi_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}
\]

For \( x = [-1, 0, 1] \) the quadratic shape functions are
Shape Functions 1D

\(N = 3\) three points: \(x = [x_1, x_2, x_3]\) and measures \(u = [u_1, u_2, u_3]\)

\[
\psi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \quad \psi_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}
\]

\[
\psi_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}
\]

Interpolation Polynomial

\(P_2(x) = u_1 \psi_1(x) + u_2 \psi_2(x) + u_3 \psi_3(x)\)

Now it is a parabola (quadratic interpolation)
Shape Functions 1D

Example:
Consider $x = [-1, 0, 1]$ and $u = [1.2, 0.7, 1.8]$. Compute $u(0.2)$?

$$
\psi_1(x) = \frac{(x - 0)(x - 1)}{(-1 - 0)(-1 - 1)} = \frac{1}{2} x(x - 1)
$$

$$
\psi_2(x) = \frac{(x + 1)(x - 1)}{(0 + 1)(0 - 1)} = -(x + 1) (x - 1)
$$

$$
\psi_3(x) = \frac{(x + 1)(x - 0)}{(1 + 1)(1 - 0)} = \frac{1}{2} x(x + 1)
$$

$$
P_2(0.2) = 1.2\psi_1(0.2) + 0.7\psi_2(0.2) + 1.8\psi_3(0.2) = 0.7920
$$
Shape Functions 1D

- Matlab code:

```matlab
x = 0:0.25:1; % measure points
f = @(x) x.^3-3*x.^2-3*sin(5*x); %inline exemple function
y = f(x); %measured values
figure()
plot(x,y,'o','Marker','o','MarkerFaceColor','red');
title('Measured values'); % plot only measured values
figure()
plot(x,y,'o','Marker','o','MarkerFaceColor','red');
hold on;
xx = 0:0.01:1; % take many points for a better plot
yy = f(xx);
plot(xx, yy,'-')
title('True function'); % plot the function
hold off
```
Shape Functions 1D

- Matlab code (continuation):

```matlab
figure() % Polyfit degree 4 function (interpolator)
plot(x,y,'o','Marker','o','MarkerFaceColor','red');
hold on;
plot(xx, yy, '-. ')
p = polyfit(x,y,4);
yyy = polyval(p, xx);
plot(xx, yyy, '-.b')
title('Degree 4 approximation');
hold off
```
Shape Functions 1D

- Interpolation Results:
**Approximation:** In case $n < (N - 1)$ there is NO solution (incompatible system) because the polynomial can not pass through all the points. In that case: \textbf{mean square approx}

$$P_n(x) = \min(\| \tilde{P}_n(x) - f(x) \|)$$
Shape Functions 1D

• **Higher interpolation:** When \( N \) is big (>8) there is a better solution than using a polynomial of high degree.

• **Runge’s Phenomenon:**

\[
f(x) = \frac{1}{1 + 25x^2}
\]

- The red curve is the Runge function \( f(x) \).
- The blue curve is a 5th-order interpolating polynomial (using six equally spaced interpolating points).
- The green curve is a 9th-order interpolating polynomial (using ten equally spaced interpolating points).
Two possible approaches:

**Polygonal curve:** Take the points in pairs and build a line segment (1st order interp) in each case. **Continuous but not derivable.**

Matlab code:

```matlab
%% Polygonal
define the function:
figure()
x=0:0.1:1;
y=f(x);
plot(x,y,'o', 'Marker','o', 'MarkerFaceColor','red');
hold on;
plot(xx,yy,'-')

interpolate the curve:
xxx=0:0.01:1;
yyy=interp1(x,y,xx);  %only substitution is possible
plot(xxx,yyy,'b')
hold off
```
Shape Functions 1D

Two possible approaches:

**Spline curve (cubic):** Take the points in groups of 2 and build a third order Polynomial (cubic interp) in each case imposing $C^2$ continuity.

Matlab code (continuation):

```matlab
%% Spline
figure()
x = 0:0.1:1;
y = f(x);
plot(x,y,'o','Marker','o','MarkerFaceColor','red');
hold on;
plot(xx,yy,'-')
xxx = 0:0.01:1;
yyy = spline(x,y,xx); %only substitution is possible
plot(xxx,yyy,'b')
hold off
```
Shape Functions 2D
2D interpolation

How can we extend the interpolation problem to 2D? That is, how can we approximate the value of $u(x, y)$ using a 2D polynomial $P_n(x, y)$?

We restrict to fixed shapes:

**Triangular element**
Numbered counterclockwise

**Quadrilateral element**
2D interpolation

From the measured values $u_i$ at the vertices we can interpolate the value of $u(x, y)$ at any other point in the triangle.

$$u(x, y) = \sum_{i=1}^{3} u_i \cdot \psi_i (x, y)$$
Triangular Shape Functions

We need the equivalent to the Lagrange polynomials (2-dim)
\[ \psi_i^e(x, y) = a_i + \beta_i x + \gamma_i y \]  
(three unknowns)
such that, it’s value is 1 for vertex \( i \), and 0 for the other two.

For the first vertex:
\[ 1 = \psi_1^e(x_1, y_1) = a_1 + \beta_1 x_1 + \gamma_1 y_1 \]
\[ 0 = \psi_1^e(x_2, y_2) = a_1 + \beta_1 x_2 + \gamma_1 y_2 \]
\[ 0 = \psi_1^e(x_3, y_3) = a_1 + \beta_1 x_3 + \gamma_1 y_3 \]

\[
\begin{pmatrix}
1 & x_1 & y_1 \\
1 & x_2 & y_2 \\
1 & x_3 & y_3
\end{pmatrix}
\begin{pmatrix}
a_1 \\
\beta_1 \\
\gamma_1
\end{pmatrix} =
\begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix} \quad \Rightarrow \quad \mathbf{c} = \mathbf{A}\backslash\mathbf{b} \quad \text{(matlab solution)}
\]
Triangular Shape Functions

Explicitly

$$\psi^e_i(x, y) = a_i + \beta_i x + \gamma_i y, \quad i = 1, 2, 3$$

$$a_i = \frac{x_j y_k - x_k y_j}{2 A_e}$$

$$\beta_i = \frac{y_j - y_k}{2 A_e}$$

$$\gamma_i = \frac{x_k - x_j}{2 A_e}$$

where $(i, j, k) = (1, 2, 3)$ cyclic and $A_e$ is the Area of $\Omega_e$

Remember:

$$\text{Area} = \frac{1}{2} \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}$$
Triangular Shape Functions

Once we have the shape functions $\psi_i^e(x, y), \quad i = 1, 2, 3$
(geometrically they are planes passing through one of the edges)

finally, the interpolation value of a point $(\tilde{x}, \tilde{y})$ is

$$
    u(\tilde{x}, \tilde{y}) = u(x_1, y_1)\psi_1^e(\tilde{x}, \tilde{y}) + u(x_2, y_2)\psi_2^e(\tilde{x}, \tilde{y}) + u(x_3, y_3)\psi_3^e(\tilde{x}, \tilde{y})
$$

The terms

$$
    \alpha_i^e = \psi_i^e(\tilde{x}, \tilde{y})
$$

are known as Barycentric Coordinates of $(\tilde{x}, \tilde{y})$
(We’ll see more in the practical sessions)
Triangular Shape Functions

Barycentric coordinates: \[ u(x, y) = \alpha_1 \cdot u_1 + \alpha_2 \cdot u_2 + \alpha_3 \cdot u_3 \]

P1: \[ 1 = \alpha_1 + \alpha_2 + \alpha_3 \]

P2: \[ \begin{pmatrix} x \\ y \end{pmatrix} = \alpha_1 \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \alpha_2 \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \alpha_3 \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \]
**Triangular Shape Functions**

**Example:** The temperature of a thin triangular plate has been measured at the vertices $T(v_1) = 40^\circ$, $T(v_2) = 90^\circ$, $T(v_3) = 10^\circ$.

If the coordinates of the vertices are: $v_1 = (0,0)$, $v_2 = (5,0)$, $v_3 = (2,3)$, estimate the temperature at the point $p = (3,1)$?

**Solution:** using the explicit formulas

$$
\psi_i^e(x, y) = a_i + \beta_i x + \gamma_i y, \quad i = 1, 2, 3
$$

$$
a_i = \frac{x_jy_k - x_ky_j}{2A_e}
$$

$$
\beta_i = \frac{y_j - y_k}{2A_e}
$$

$$
\gamma_i = \frac{x_k - x_j}{2A_e}
$$

For the first shape function, we have:

$$
A_e = \frac{15}{2}
$$

$$
a_1 = \frac{x_2y_3 - x_3y_2}{2A_e} = 1
$$

$$
\beta_1 = \frac{y_2 - y_3}{2A_e} = \frac{-1}{5}
$$

$$
\gamma_1 = \frac{x_3 - x_2}{2A_e} = \frac{-1}{5}
$$

First Shape Function

$$
\psi_1^e(x, y) = 1 - x \frac{3}{5} - y \frac{3}{5}
$$
Triangular Shape Functions

**Example:** The temperature of a thin triangular plate has been measured at the vertices $T(v_1) = 40^\circ$, $T(v_2) = 90^\circ$, $T(v_3) = 10^\circ$.

If the coordinates of the vertices are: $v_1 = (0,0)$, $v_2 = (5,0)$, $v_3 = (2,3)$, estimate the temperature at the point $p = (3,1)$?

**Solution:** using the explicit formulas

$$\psi_i^e(x, y) = a_i + \beta_i x + \gamma_i y, \quad i = 1, 2, 3$$

$$a_i = \frac{x_j y_k - x_k y_j}{2A_e}$$

$$\beta_i = \frac{y_j - y_k}{2A_e}$$

$$\gamma_i = \frac{x_k - x_j}{2A_e}$$

For $i = 2, j = 3, k = 1$:

$$A_e = \frac{15}{2}$$

$$a_2 = \frac{x_3 y_1 - x_1 y_3}{2A_e} = 0$$

$$\beta_2 = \frac{y_3 - y_1}{2A_e} = \frac{1}{5}$$

$$\gamma_2 = \frac{x_1 - x_3}{2A_e} = -\frac{2}{15}$$

$$\psi_2^e(x, y) = 1 + \frac{x}{5} - \frac{2y}{15}$$

Second Shape Function
Triangular Shape Functions

**Example:** The temperature of a thin triangular plate has been measured at the vertices $T(v_1) = 40^\circ$, $T(v_2) = 90^\circ$, $T(v_3) = 10^\circ$. If the coordinates of the vertices are: $v_1 = (0,0)$, $v_2 = (5,0)$, $v_3 = (2,3)$, estimate the temperature at the point $p = (3,1)$?

**Solution:** using the explicit formulas

$$\psi_i^e(x, y) = a_i + \beta_i \, x + \gamma_i \, y, \quad i = 1, 2, 3$$

- $a_i = \frac{x_j y_k - x_k y_j}{2A_e}$
- $\beta_i = \frac{y_j - y_k}{2A_e}$
- $\gamma_i = \frac{x_k - x_j}{2A_e}$

$i = 3, j = 1, k = 2$

$$A_e = \frac{15}{2}$$

$$a_3 = \frac{x_1 y_2 - x_2 y_1}{2A_e} = 0$$

$$\beta_3 = \frac{y_1 - y_2}{2A_e} = 0$$

$$\gamma_3 = \frac{x_2 - x_1}{2A_e} = \frac{1}{3}$$

$$\psi_3^e(x, y) = 0 + 0 \cdot x + \frac{y}{3}$$

Third Shape Function
Triangular Shape Functions

**Example:** The temperature of a thin triangular plate has been measured at the vertices $T(v_1) = 40^\circ$, $T(v_2) = 90^\circ$, $T(v_3) = 10^\circ$.

If the coordinates of the vertices are: $v_1 = (0,0)$, $v_2 = (5,0)$, $v_3 = (2,3)$, estimate the temperature at the point $p = (3,1)$?

**Solution:**

\[
\psi_1^e(x,y) = 1 - \frac{x}{5} - \frac{y}{5} \quad \rightarrow \quad \alpha_1 = \psi_1^e(3,1) \\
\psi_2^e(x,y) = 1 + \frac{x}{5} - \frac{2y}{15} \quad \rightarrow \quad \alpha_2 = \psi_2^e(3,1) \\
\psi_3^e(x,y) = 0 - 0x + \frac{y}{3} \quad \rightarrow \quad \alpha_3 = \psi_3^e(3,1) \\
\]

\[
T(p) = 40\alpha_1 + 90\alpha_2 + 10\alpha_3 = 53.3333^\circ 
\]

**Exercise:** Compute the shape functions associated to the triangle $v_1 = (1,1)$, $v_2 = (3,2)$, $v_3 = (2,4)$. \[
\psi_1^e(x,y) = \frac{1}{5}(8 - 2x - y) \\
\psi_2^e(x,y) = \frac{1}{5}(-2 + 3x - y) \\
\psi_3^e(x,y) = \frac{1}{5}(-1 - x + 2y) 
\]
Quadrilateral Shape Functions

A similar situation is found when a **rectangular** element is used.

\[ \psi_i^e (x, y) = c_1 + c_2 x + c_3 y + c_4 xy \] (four unknowns)

such that, it’s value is 1 for vertex \( i \), and 0 for the other three:

For the first vertex:

\[
1 = \psi_1^e (x_1, y_1) = c_1 + c_2 x_1 + c_3 y_1 + c_4 x_1 y_1 \\
0 = \psi_1^e (x_2, y_2) = c_1 + c_2 x_2 + c_3 y_2 + c_4 x_2 y_2 \\
0 = \psi_1^e (x_3, y_3) = c_1 + c_2 x_3 + c_3 y_3 + c_4 x_3 y_3 \\
0 = \psi_1^e (x_4, y_4) = c_1 + c_2 x_4 + c_3 y_4 + c_4 x_4 y_4
\]

Analogously for the other vertices

(This is not correct for general quadrilateral elements)
Quadrilateral Shape Functions

General expressions for **Rectangles:**

If we denote by

\[ a = x_2 - x_1 \]
\[ b = y_2 - y_1 \]

Then

\[ \psi_1(x, y) = \frac{1}{ab} (x - x_2)(y - y_2) \]
\[ \psi_2(x, y) = \frac{-1}{ab} (x - x_1)(y - y_2) \]
\[ \psi_3(x, y) = \frac{1}{ab} (x - x_1)(y - y_1) \]
\[ \psi_4(x, y) = \frac{-1}{ab} (x - x_2)(y - y_1) \]
Quadrilateral Shape Functions

The **reference** bilinear quadrilateral element \( \Omega^R: [-1,1] \times [-1,1] \)

We can use matlab to compute the coeff.

\[
\psi^e_j(x_i, y_i) = c_1 j + c_2 j x_i + c_3 j y_i + c_4 j x_i y_i = \delta_{ij}
\]

\[
\text{punts} = \begin{bmatrix}
-1, -1; \\
1, -1; \\
1, 1; \\
-1, 1;
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
\text{ones}(4,1), \text{punts}(:,1), \text{punts}(:,2), \text{punts}(:,1).*\text{punts}(:,2)
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
1; 0; 0; 0;
\end{bmatrix}
\]

\[
C_1 = A \backslash b \quad \% \text{coeff of the first shape function}
\]

\% **exercise**: do it for the other shape functions
Quadrilateral Shape Functions

The shape functions for the reference element $\Omega^R$, are:

\[
\begin{align*}
\psi_1^R (\xi, \eta) &= \frac{(1 - \xi)(1 - \eta)}{2}, \\
\psi_2^R (\xi, \eta) &= \frac{(1 + \xi)(1 - \eta)}{2}, \\
\psi_3^R (\xi, \eta) &= \frac{(1 + \xi)(1 + \eta)}{2}, \\
\psi_4^R (\xi, \eta) &= \frac{(1 - \xi)(1 + \eta)}{2}.
\end{align*}
\]

Or equivalently:

\[
\begin{align*}
\psi_1^R (\xi, \eta) &= \frac{1 - \xi - \eta + \xi \eta}{4}, \\
\psi_2^R (\xi, \eta) &= \frac{1 + \xi - \eta - \xi \eta}{4}, \\
\psi_3^R (\xi, \eta) &= \frac{1 + \xi + \eta + \xi \eta}{4}, \\
\psi_4^R (\xi, \eta) &= \frac{1 - \xi + \eta - \xi \eta}{4}.
\end{align*}
\]
Quadrilateral Shape Functions

Example: The temperature of at the vertices of the reference quadrilateral has been measured as $T(v_1) = 10^\circ$, $T(v_2) = 20^\circ$, $T(v_3) = 30^\circ$, $T(v_4) = 40^\circ$.

Estimate the temperature at the point $p = (-0.3,0.5)$?

Solution:

$$
\psi_1^R(p) = 0.1625; \quad \psi_2^R(p) = 0.0875;
$$

$$
\psi_3^R(p) = 0.2625; \quad \psi_4^R(p) = 0.4875;
$$

$$
T(p) = T_1\psi_1^R(p) + T_2\psi_2^R(p) + T_3\psi_3^R(p) + T_4\psi_4^R(p) = 30.75^\circ
$$
Quadrilateral Shape Functions

For a General Quadrilateral element usually the shape function is NOT a 2 degree polynomial. Because of that it is not easy to compute these functions, but we still can compute the Barycentric Coordinates.

How to compute $\alpha_i^k = \psi_i^k (P)$?
Quadrilateral Shape Functions

• Change from the Reference quadrilateral to \([0,1] \times [0,1]\)

\[
\lambda = \frac{\xi + 1}{2}, \quad \mu = \frac{\eta + 1}{2}
\]

**Shape functions** on \([0,1] \times [0,1]\):

\[
\psi_1(\lambda, \mu) = (1 - \lambda)(1 - \mu) \quad \psi_3(\lambda, \mu) = \lambda\mu
\]

\[
\psi_2(\lambda, \mu) = \lambda(1 - \mu) \quad \psi_4(\lambda, \mu) = (1 - \lambda)\mu
\]
Quadrilateral Shape Functions

- **General Quadrilateral:** *Isoparametric* transformation

\[
(x, y) = \psi_1(\lambda, \mu)v_1 + \psi_2(\lambda, \mu)v_2 + \psi_3(\lambda, \mu)v_3 + \psi_4(\lambda, \mu)v_4
\]

Change to another quadrilateral using the shape functions: for simplicity we will use here the rectangle [0,1]x[0,1]:

\[
P(\lambda, \mu) = (1 - \lambda)(1 - \mu) v_1 + \lambda(1 - \mu) v_2 + \lambda\mu v_3 + (1 - \lambda)\mu v_4
\]

\[
\begin{align*}
\alpha_1 & = (1 - \lambda)(1 - \mu) \\
\alpha_2 & = \lambda(1 - \mu) \\
\alpha_3 & = \lambda\mu \\
\alpha_4 & = (1 - \lambda)\mu
\end{align*}
\]
Quadrilateral Shape Functions

We can compute them using $\lambda, \mu \in [0,1]$, as a parametrization of the quadrilateral edges.

$$ P = (1 - \lambda)(1 - \mu) v_1 + \lambda(1 - \mu) v_2 + \lambda \mu v_3 + (1 - \lambda)\mu v_4 $$

Rearranging the terms, the previous equation can be written as:

$$ a + \lambda b + \mu c + \lambda \mu d = 0 $$

where

$$ a = v_1 - P, \quad b = v_2 - v_1, \quad c = v_4 - v_1, \quad d = v_1 - v_2 + v_3 - v_4 $$
Quadrilateral Shape Functions

In fact, we have to solve a system of two non-linear equations. We will use Newton’s iterative method:

Our system is:

\[
\begin{pmatrix}
    a_x \\
    a_y
\end{pmatrix} + \lambda \begin{pmatrix}
    b_x \\
    b_y
\end{pmatrix} + \mu \begin{pmatrix}
    c_x \\
    c_y
\end{pmatrix} + \lambda \mu \begin{pmatrix}
    d_x \\
    d_y
\end{pmatrix} = \begin{pmatrix}
    0 \\
    0
\end{pmatrix}
\]

or

\[
\begin{cases}
    f(\lambda, \mu) = 0 \\
    g(\lambda, \mu) = 0
\end{cases}
\]

Newton’s method

\[
\begin{pmatrix}
    \lambda \\
    \mu
\end{pmatrix}_{n+1} = \begin{pmatrix}
    \lambda \\
    \mu
\end{pmatrix}_n - \begin{pmatrix}
    \frac{\partial f}{\partial \lambda}(\lambda_n, \mu_n) & \frac{\partial f}{\partial \mu}(\lambda_n, \mu_n) \\
    \frac{\partial g}{\partial \lambda}(\lambda_n, \mu_n) & \frac{\partial g}{\partial \mu}(\lambda_n, \mu_n)
\end{pmatrix}^{-1} \begin{pmatrix}
    f(\lambda_n, \mu_n) \\
    g(\lambda_n, \mu_n)
\end{pmatrix}
\]
Quadrilateral Shape Functions

In our case:
\[
\frac{\partial f}{\partial \lambda} = b_x + \mu d_x \quad \frac{\partial f}{\partial \mu} = c_x + \lambda d_x \\
\frac{\partial g}{\partial \lambda} = b_y + \mu d_y \quad \frac{\partial g}{\partial \mu} = c_y + \lambda d_y
\]

then
\[
\begin{pmatrix}
\lambda \\
\mu
\end{pmatrix}_{n+1} = \begin{pmatrix}
\lambda \\
\mu
\end{pmatrix}_n - \begin{pmatrix}
\frac{b_x + \mu_n d_x}{c_x + \lambda_n d_x} & \frac{b_x + \mu_n d_x}{c_x + \lambda_n d_x} \\
\frac{b_y + \mu_n d_y}{c_y + \lambda_n d_y} & \frac{b_y + \mu_n d_y}{c_y + \lambda_n d_y}
\end{pmatrix}^{-1} \begin{pmatrix}
f(\lambda_n, \mu_n) \\
g(\lambda_n, \mu_n)
\end{pmatrix}
\]

Or
\[
\begin{pmatrix}
b_x + \mu_n d_x & c_x + \lambda_n d_x \\
b_y + \mu_n d_y & c_y + \lambda_n d_y
\end{pmatrix} \begin{pmatrix}
\Delta \lambda \\
\Delta \mu
\end{pmatrix} = -\begin{pmatrix}
f(\lambda_n, \mu_n) \\
g(\lambda_n, \mu_n)
\end{pmatrix}
\]

\[
\begin{pmatrix}
\lambda \\
\mu
\end{pmatrix}_{n+1} = \begin{pmatrix}
\lambda \\
\mu
\end{pmatrix}_n + \begin{pmatrix}
\Delta \lambda \\
\Delta \mu
\end{pmatrix}
\]
Quadrilateral Shape Functions

- Example: Given a quadrilateral defined by vertices $v_1 = (0,0), v_2 = (5,-1), v_3 = (4,5), v_4 = (1,4)$ compute the barycentric coordinates of point $P=(3,2)$.

Compute: $\mathbf{a} + \lambda \mathbf{b} + \mu \mathbf{c} + \lambda \mu \mathbf{d} = 0$

where $\mathbf{a} = v_1 - P$, $\mathbf{b} = v_2 - v_1$, $\mathbf{c} = v_4 - v_1$, $\mathbf{d} = v_1 - v_2 + v_3 - v_4$.

\[
\begin{pmatrix}
5 + \mu_n (-1) & 1 + \lambda_n (-1) \\
-1 + \mu_n 2 & 4 + \lambda_n 2
\end{pmatrix}
\begin{pmatrix}
\Delta \lambda \\
\Delta \mu
\end{pmatrix}
= -
\begin{pmatrix}
f (\lambda_n, \mu_n) \\
g (\lambda_n, \mu_n)
\end{pmatrix}
\]

Initialization:

\[
\begin{pmatrix}
\lambda_0 \\
\mu_0
\end{pmatrix}
= \begin{pmatrix}
0.1 \\
0.1
\end{pmatrix}
\]

can be any value between 0 and 1.

Sol. $\lambda = 0.6250, \mu = 0.5$

$\alpha = [0.1875 \ 0.3125 \ 0.3125 \ 0.1875]$