Mesh structure

To set up the finite element problem, the table of coordinates and connectivity need to be set. Since only one element is considered, the meash is really simple

1. Nodal coordinates

Global node number	x	y
1	0	0
2	L	0

2. Connectivity table

Element	**Left node (1)**	**Right Node (2)**
1	1	2

In order to find the equivalent node at the extremities, the virtual work principle is invoked. The virtual work performed by the external load is given by

$$\delta W = \int_0^L \delta v \, p \, dx$$

The displacement is computed via the finite element expansion $v = \mathbf{N}\mathbf{v}$ where

$$N = \begin{pmatrix} N_1 & N_2 & N_3 & N_4 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} v_1 & \phi_1 & v_2 & \phi_2 \end{pmatrix}^\top .$$

Plugging the approximation in the virtual work it is obtained

$$\delta W = \delta \mathbf{v}^\top \int_0^L \mathbf{N}^\top \, p \, dx = \delta \mathbf{v}^\top \mathbf{f}.$$

So the expression of the generalized force is given by

$$\mathbf{f} = \int_0^L \mathbf{N}^\top \, p \, dx$$

Developing the computations

$$
\begin{aligned}
f_1 &= \int_0^L \left(1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right)\left(p_1 + (p_2 - p_1)\frac{x}{L}\right)\,dx = 0,\\
f_2 &= \int_0^L \left(x - \frac{2x^2}{L} + \frac{x^3}{L^2}\right)\left(p_1 + (p_2 - p_1)\frac{x}{L}\right)\,dx,\\
f_3 &= \int_0^L \left(\frac{3x^2}{L^2} - \frac{2x^3}{L^3}\right)\left(p_1 + (p_2 - p_1)\frac{x}{L}\right)\,dx,\\
f_4 &= \int_0^L \left(-\frac{x^2}{L} + \frac{x^3}{L^2}\right)\left(p_1 + (p_2 - p_1)\frac{x}{L}\right)\,dx.
\end{aligned}
$$

This computations can be performed in python using the sympy librairy and rescaling the integral to 1 by intrucing $\xi = x/L$

In [1]:
```
import sympy as sp

p_1 = sp.symbols('p_1')
p_2 = sp.symbols('p_2')
L = sp.symbols('L')
xi = sp.symbols('xi')

load = p_1 + (p_2 - p_1) * xi
jacobian = L

N_1 = 1 - 3 * xi**2 + 2*xi**3
N_2 = (xi - 2*xi**2 + xi**3)*L
N_3 = 3*xi**2 - 2*xi**3
N_4 = (-xi**2 + xi**3)*L

N_vec = sp.Matrix([N_1, N_2, N_3, N_4])

f = sp.integrate(N_vec*load*jacobian, (xi, 0, 1))

f
```

Out[1]:

$$
\begin{bmatrix} \frac{7Lp_1}{20} + \frac{3Lp_2}{20} \\ \frac{L^2p_1}{20} + \frac{L^2p_2}{30} \\ \frac{3Lp_1}{20} + \frac{7Lp_2}{20} \\ -\frac{L^2p_1}{30} - \frac{L^2p_2}{20} \end{bmatrix}
$$

If the load is constant, the result is given by

In [11]:
```
p = sp.symbols('p')
f = sp.integrate(N_vec*p*jacobian, (xi, 0, 1))
f
```

Out[11]: $\left[\begin{matrix}\frac{L p}{2}\\\frac{L^{2} p}{12}\\\frac{L p}{2}\\- \frac{L^{2} p}{12}\end{matrix}\right]$