# Element Types

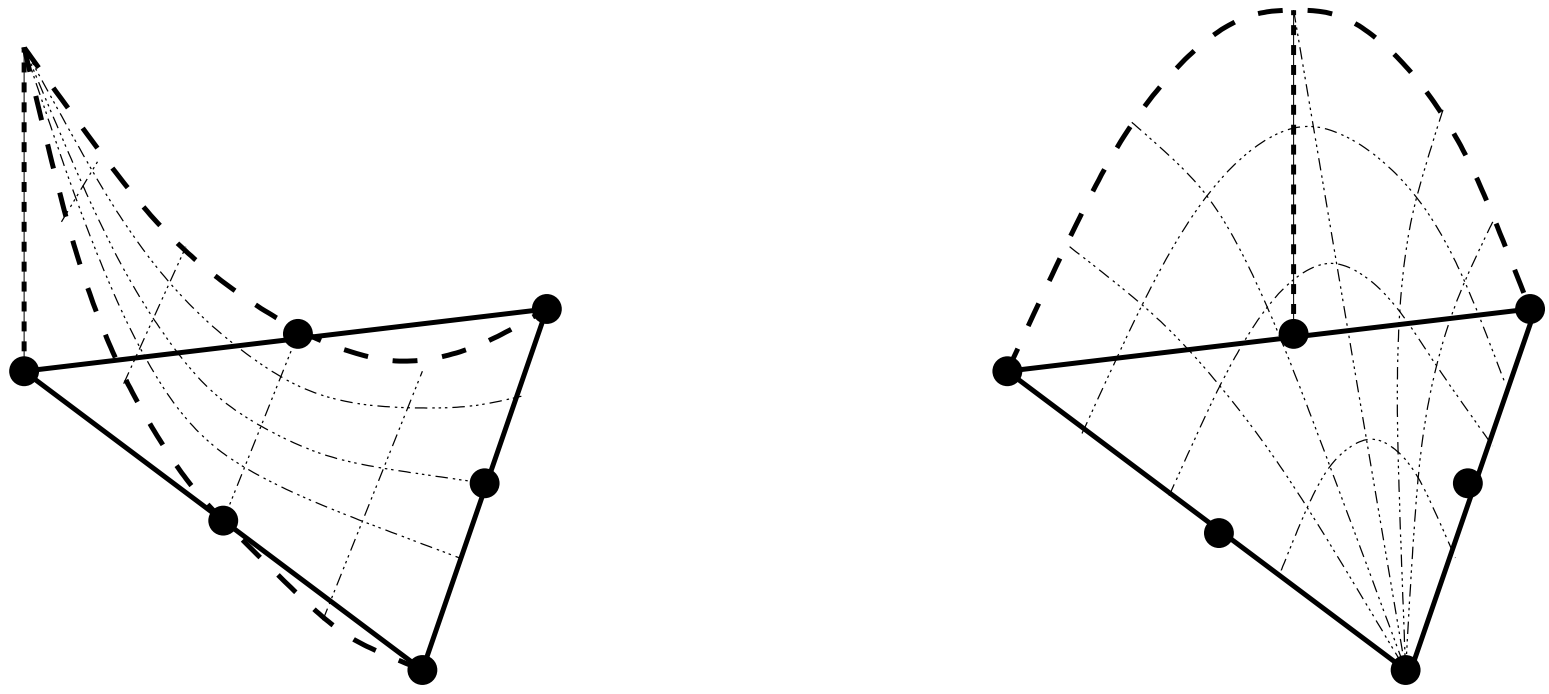
Our description so far has concerned only piecewise linear ($P_1$) basis functions ('hat' functions in 1-dimension and 'tent' functions in 2-dimensions) and correspondingly only piecewise linear finite element approximations $u_h$, but there are many other useful approximation spaces with convenient local bases for which the implementation we have described is similar.

On triangles, we can define a piecewise quadratic (or $P_2$) approximation space by having nodes at vertices and also at all midpoints of sides. By insisting that there is one basis function $\phi_j$ for each node $j = 1, 2, \ldots, n + n_\partial$ which satisfies

$$\phi_j(\text{node } i) = \delta_{i,j}$$

we can compute the piecewise quadratic Galerkin finite element approximation $u_h$.

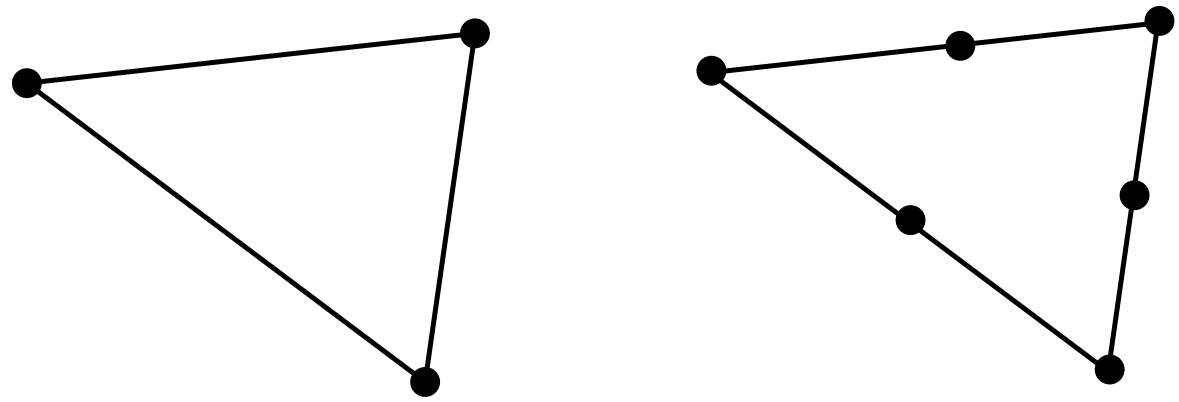
The basis functions $\phi_j, j = 1, 2, \ldots, n$ are continuous, have support only on the patch of elements around node $j$ and on a single element look like:



Note there are two types: those being 1 at a vertex and those being 1 at a midside. Both satisfy $\phi(\text{node } i) = \delta_{i,j}$

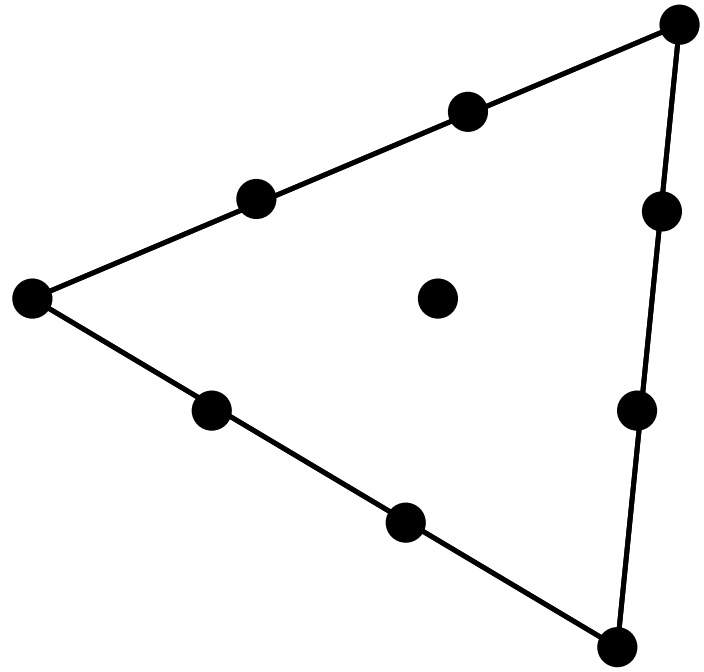
Note that $P_2$ approximation defined in term of these basis functions is certainly continuous, and so $u_h \in \mathcal{H}_E^1$, but is not more smooth than this.

A convenient representation of $P_1$ and $P_2$ triangular elements is via the illustration:



The 3 nodes in a $P_1$ element are sufficient to uniquely determine the 3 coefficients in a bivariate linear function $ax + by + c$ and the 6 nodes in a $P_2$ element are sufficient to uniquely determine the 6 coefficients in a bivariate quadratic function $ax^2 + bxy + cy^2 + dx + ey + f$.

In general triangular $P_k$ elements are defined by including more nodes on element sides and also some internal nodes, so eg. to uniquely define bivariate cubics, the $P_3$ element has 10 nodes:



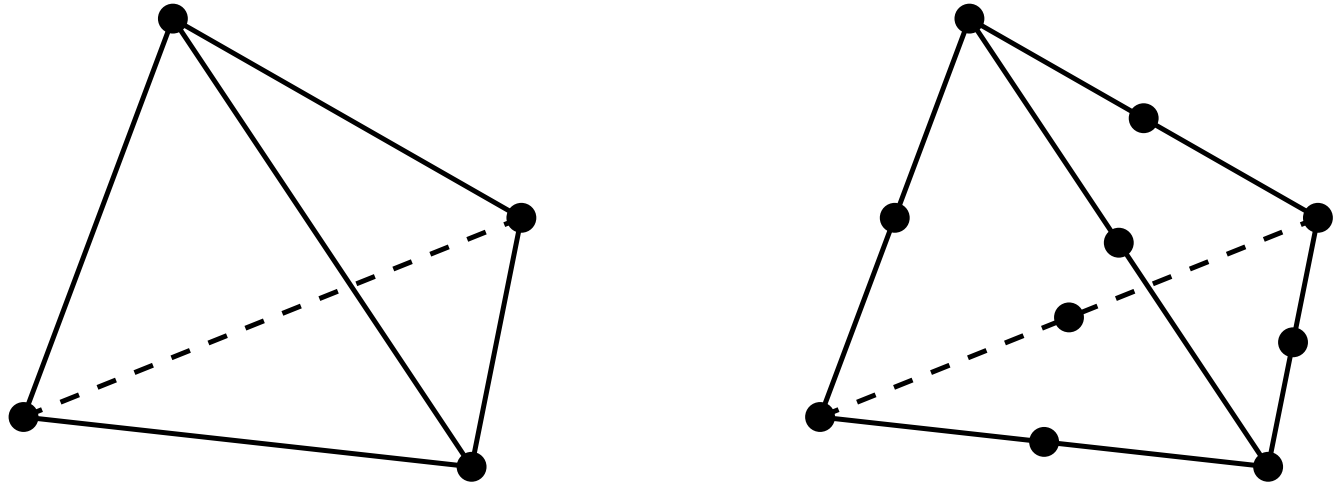
On the reference element the internal node at $(\frac{1}{3}, \frac{1}{3})$ corresponds to the basis function $27\,\xi\,\eta\,(1-\xi-\eta)$ for example

If $\Omega \subset \mathbb{R}^3$, then $P_k$ approximation is defined on a tesselation of tetrahedral (or simplex) elements. As in 2-dimensions, the tetrahedra must fill the domain, not overlap and edges must meet at common nodes.

For $P_1$, $u_h$ will be linear on each tetrahedral element ie. of the form $ax + by + cz + d$ and so with the 4 vertices as nodes the 4 element basis functions
$\psi_{k,j}(x, y, z), j = 1, 2, 3, 4$ are readily defined to satisfy
$\psi_{k,j}(\text{vertex } i) = \delta_{i,j}$, hence defining continuous piecewise linear global basis functions with support on a patch of tetrahedra surrounding a node and satisfying

$$\phi_j(\text{node } i) = \begin{cases} 1 & \text{when } i = j, \\ 0 & \text{when } i \neq j \ . \end{cases}$$

We represent $P_1$ and $P_2$ elements for example by the illustration



# Quadrilateral Elements

It is clear than any polygonal domain can be tesselated with triangles and so $P_k$ elements provide a convenient and flexible approximation space.
There are certain different advantages however to the use of rectangular or quadrilateral tesselations and these are quite widely used.

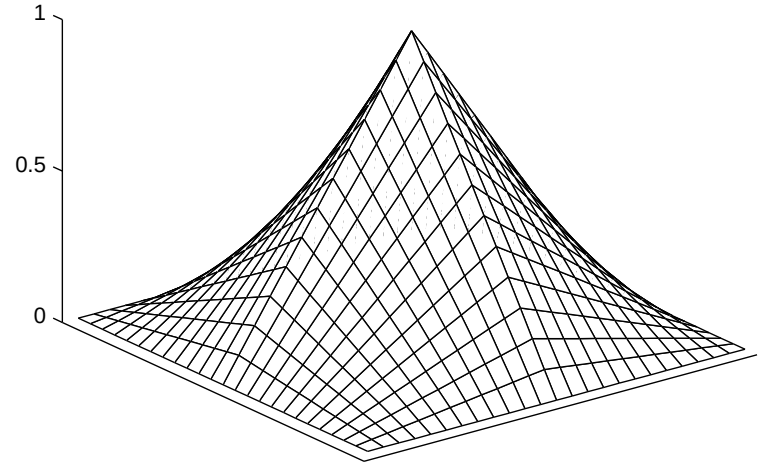
The simplest is the *bilinear* or $Q_1$ element which is of the form $(ax+b)(cy+d) = \alpha xy + \beta x + \gamma y + \delta$ on any rectangular element. Thus for example for a square element with vertices

$$(0,0), \quad (h,0), \quad (h,h), \quad (0,h)$$

the element basis functions are

$$(1-x/h)(1-y/h), \quad x/h(1-y/h), \quad xy/h^2, \quad (1-x/h)y/h$$

The global basis functions $\phi_j(x, y)$ satisfy $\phi_j(\text{node } i) = \delta_{i,j}$ as usual and on a regular mesh of square elements look like




Note that $Q_1$ basis functions have some curvature because of the 'twist' term $xy$ which is not present with $P_1$ elements. For a rectangular coordinate-aligned mesh they are linear on each edge, hence continuous across each edge which ensures that $u_h \in \mathcal{H}^1_E$.

For general quadrilateral elements (with non-coordinate aligned edges) the isoparametric bilinear mapping is *necessary* to ensure that $Q_1$ approximation is continuous, hence

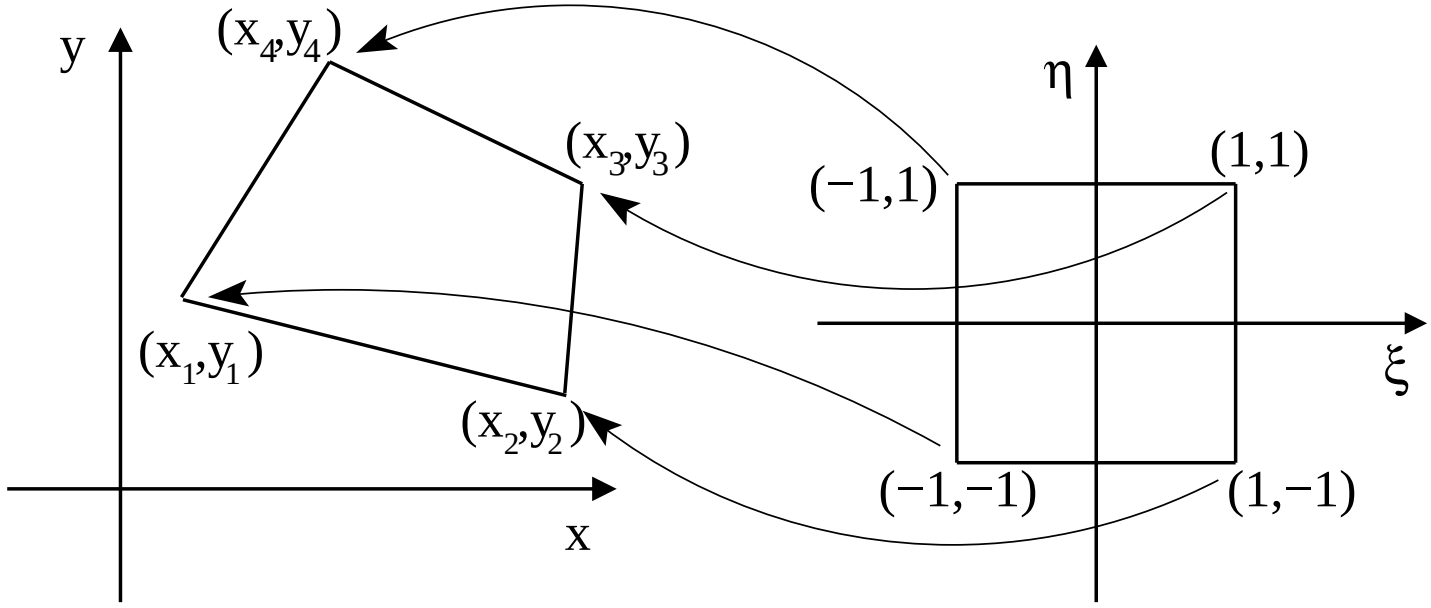
$$u_h = \sum_{j=1}^{n+n_\partial} \mathbf{u}_j \phi_j \in \mathcal{H}_E^1$$

On the reference element $\square_*$, $(\xi, \eta) \in [-1,1] \times [-1,1]$ we define the element basis functions by

$$\begin{aligned}
\chi_1(\xi,\eta) &= (\xi-1)(\eta-1)/4 \\
\chi_2(\xi,\eta) &= -(\xi+1)(\eta-1)/4 \\
\chi_3(\xi,\eta) &= (\xi+1)(\eta+1)/4 \\
\chi_4(\xi,\eta) &= -(\xi-1)(\eta+1)/4
\end{aligned}$$

Then the isoparametric mapping is

$$x(\xi,\eta)=\sum_{\nu=1}^{4} x_\nu \chi_\nu(\xi,\eta), \quad y(\xi,\eta)=\sum_{\nu=1}^{4} y_\nu \chi_\nu(\xi,\eta)$$




The important point here is that the mapped element basis function defined on a general quadrilateral element is linear along each edge which ensures continuity across edges and so guarentees $\mathcal{H}^1_E$ approximation

The Jacobian for this bilinear isoparametric transformation is

$$\boldsymbol{J_k} = \frac{\boldsymbol{\partial}(\boldsymbol{x},\boldsymbol{y})}{\boldsymbol{\partial}(\boldsymbol{\xi},\boldsymbol{\eta})} = \begin{bmatrix} \sum_{j=1}^{4} \boldsymbol{x_j} \frac{\boldsymbol{\partial}\boldsymbol{\chi_j}}{\boldsymbol{\partial}\boldsymbol{\xi}} & \sum_{j=1}^{4} \boldsymbol{y_j} \frac{\boldsymbol{\partial}\boldsymbol{\chi_j}}{\boldsymbol{\partial}\boldsymbol{\xi}} \\ \sum_{j=1}^{4} \boldsymbol{x_j} \frac{\boldsymbol{\partial}\boldsymbol{\chi_j}}{\boldsymbol{\partial}\boldsymbol{\eta}} & \sum_{j=1}^{4} \boldsymbol{y_j} \frac{\boldsymbol{\partial}\boldsymbol{\chi_j}}{\boldsymbol{\partial}\boldsymbol{\eta}} \end{bmatrix}$$

and the entries of the element stiffness matrix are

$$\boldsymbol{a_{ij}^{(k)}} = \int_{\square *} \left\{ \frac{\boldsymbol{\partial}\boldsymbol{\psi}_{*,\boldsymbol{i}}}{\boldsymbol{\partial}\boldsymbol{x}} \frac{\boldsymbol{\partial}\boldsymbol{\psi}_{*,\boldsymbol{j}}}{\boldsymbol{\partial}\boldsymbol{x}} + \frac{\boldsymbol{\partial}\boldsymbol{\psi}_{*,\boldsymbol{i}}}{\boldsymbol{\partial}\boldsymbol{y}} \frac{\boldsymbol{\partial}\boldsymbol{\psi}_{*,\boldsymbol{j}}}{\boldsymbol{\partial}\boldsymbol{y}} \right\} |\boldsymbol{J_k}| \, \mathrm{d}\boldsymbol{\xi}\mathrm{d}\boldsymbol{\eta},$$

$\boldsymbol{i,j} = \mathbf{1,2,3,4}$ similarly as for triangles

Quadrature (on $\square_*$) would usually be used for the load vector but also it is often used to compute (or at least approximately compute) the entries of the stiffness matrix here.

Tensor products of Gaussian quadrature rules such as

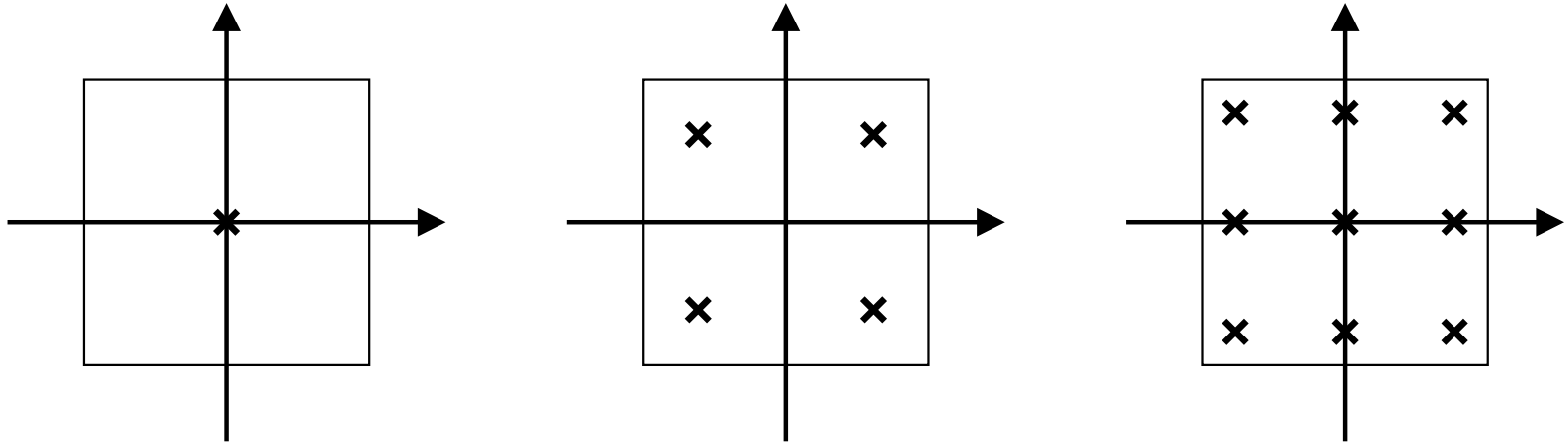
$$\int_{-1}^{1} r(\xi)\mathrm{d}\xi \approx r\left(\frac{1}{\sqrt{3}}\right) + r\left(-\frac{1}{\sqrt{3}}\right)$$

are most often used ie. for this rule

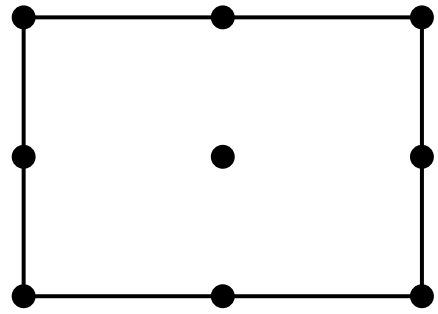
$$\int_{-1}^{1}\int_{-1}^{1} r(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \approx r\left(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right) + r\left(-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$$

$$+r\left(\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}\right) + r\left(-\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}\right)$$

# Gauss quadrature points:



Similarly as for triangles there are higher order quadrilateral elements $Q_k$, $Q_2$ being the *biquadratic* element illustrated by



which is of the form

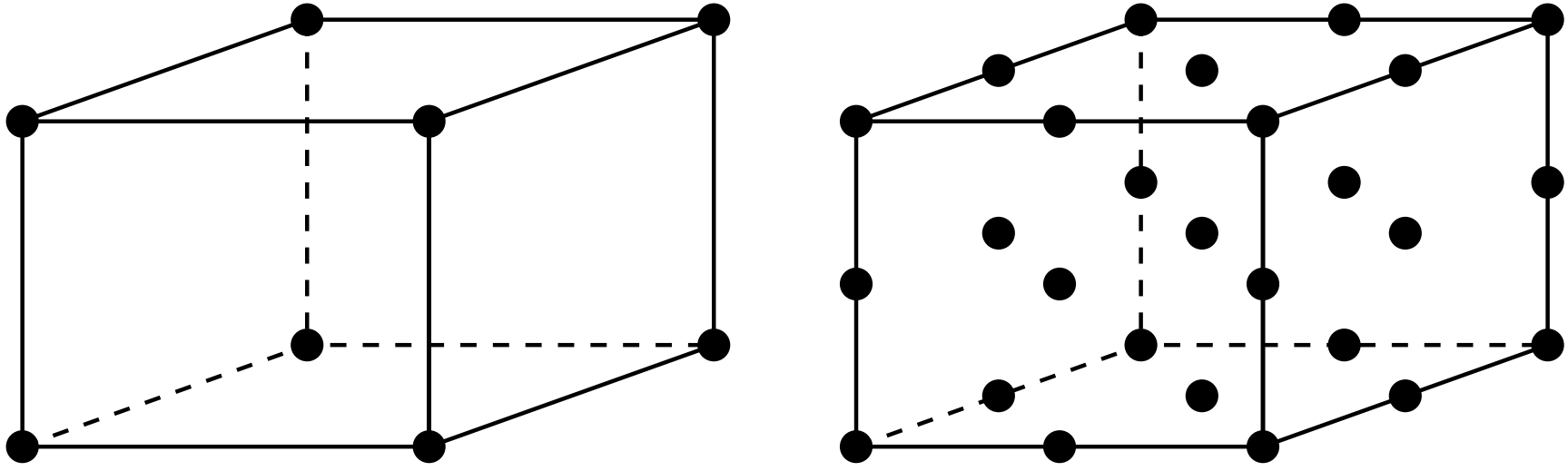
$$(ax^2 + bx + c)(dy^2 + ey + f)$$

(hence 9 coefficients of $\{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\}$) on rectangular elements

For quadrilateral elements, the parametric mapping exactly as used for $Q_1$ is employed (this is thus called a *subparametric* mapping here) and continuity across element edges is thus ensured.

For $\Omega \subset \mathbb{R}^3$ there are correspondingly $Q_k$ *brick* elements eg. 3-dimensional $Q_1$ elements (*trilinear* elements) which are of the form $(ax+b)(cy+d)(ez+f)$.

$Q_1$ and $Q_2$ elements are illustrated thus:



In general the use of a finite element basis which ensures that $u_h \in \mathcal{H}^1_E$ for the Poisson or other 2nd order elliptic partial differential equations is decribed as a *conforming* finite element approximation.

*Non-conforming* approximations such as discontinuous elements which are not smooth enough will not be considered in this course, though some such methods are used (generally for problems other than elliptic problems). Some of the implementational details for example remain similar.