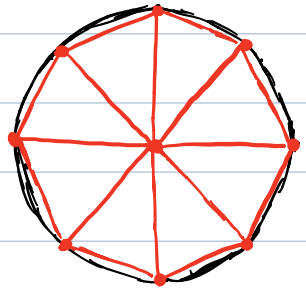
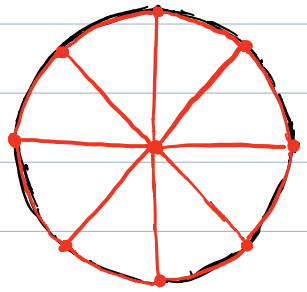
2D Heat Conduction: Isoparametric Triangular Finite Element Approximations:

In the last lecture, we discussed how to construct affine triangular finite element approximations of steady two-dimensional heat conduction. While simple and powerful, affine triangular finite element approximations suffer from a major flaw - they approximate the domain geometry using straight-sided triangles:



Unfortunately, this geometry approximation can pollute the accuracy of a finite element approximation. In fact, it limits the rate of convergence of an affine triangular finite element approximation to second-order in the H^1-norm. To overcome this geometry approximation, one can instead using a two-dimensional mesh composed of curved elements:



One way to arrive at such a mesh is to use finite element functions defined on the original triangular finite element mesh. This is the basis of so-called isoparametric triangular finite element approximations.

To proceed, let $\Delta\mathcal{M}^h = \{\Delta\Omega^e\}_{e=1}^{n_{el}}$ be a conforming triangular finite element mesh as described in the previous lecture. We denote all quantities on the mesh using a "Δ". For example, we denote the finite element approximation space of continuous piecewise polynomials of degree k with respect to the mesh as:

$$\Delta P^k_{cont.}(\Delta\mathcal{M}^h)$$

We denote the nodes on the mesh as:

$$\{\Delta\vec{X}_A\}_{A=1}^{n_{nod}}$$

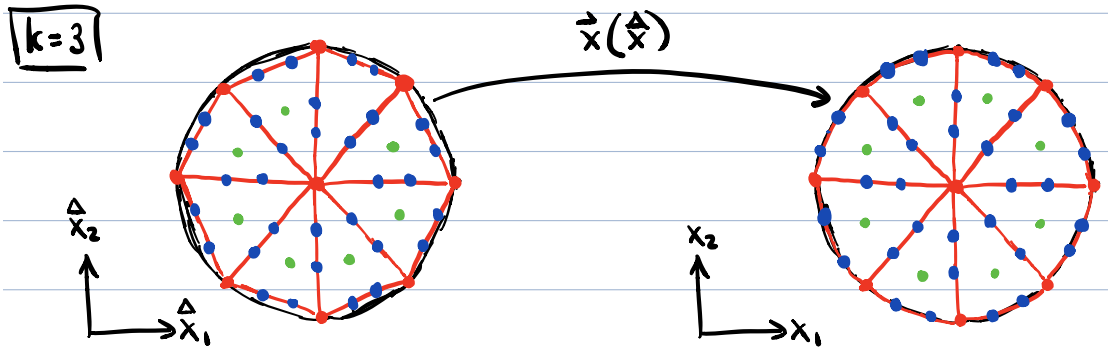
and we denote the finite element basis functions as:

$$\{\Delta N_A\}_{A=1}^{n_{nod}}$$

We can then deform the mesh using a finite element mapping:

$$\vec{x}(\Delta\vec{X}) = \sum_{A=1}^{n_{nod}} \vec{x}_A \Delta N_A(\Delta\vec{X})$$

where $\Delta\vec{X}$ is any point in $\Delta\Omega^h$. Visually:



To ensure that no element in the triangular mesh is turned inside out under the finite element mapping, we require the Jacobian determinant of the mapping be everywhere positive:

$$\det \begin{bmatrix} \frac{\partial x_1}{\partial \hat{x}_1} & \frac{\partial x_1}{\partial \hat{x}_2} \\ \frac{\partial x_2}{\partial \hat{x}_1} & \frac{\partial x_2}{\partial \hat{x}_2} \end{bmatrix} > 0$$

Then, each element $\hat{\Omega}^e$ in the triangular mesh is mapped to a curved element:

$$\Omega^e := \{ \vec{x}(\hat{x}) : \hat{x} \in \hat{\Omega}^e \}$$

and since the basis functions $\{\hat{N}_A\}_{A=1}^{n_{nod}}$ are interpolatory at the nodes $\{\hat{x}_A\}_{A=1}^{n_{nod}}$, the vertex, edge, and element nodes of each element $\hat{\Omega}^e$ are mapped to vertex, edge, and element nodes on element Ω^e associated with the mapping degrees of freedom $\{\vec{x}_A\}_{A=1}^{n_{nod}}$:

$$\vec{x}(\hat{x}_a^e) = \vec{x}_a^e$$

for $e = 1, \ldots, n_{el}$ and $a = 1, \ldots, n_{en}$ where:

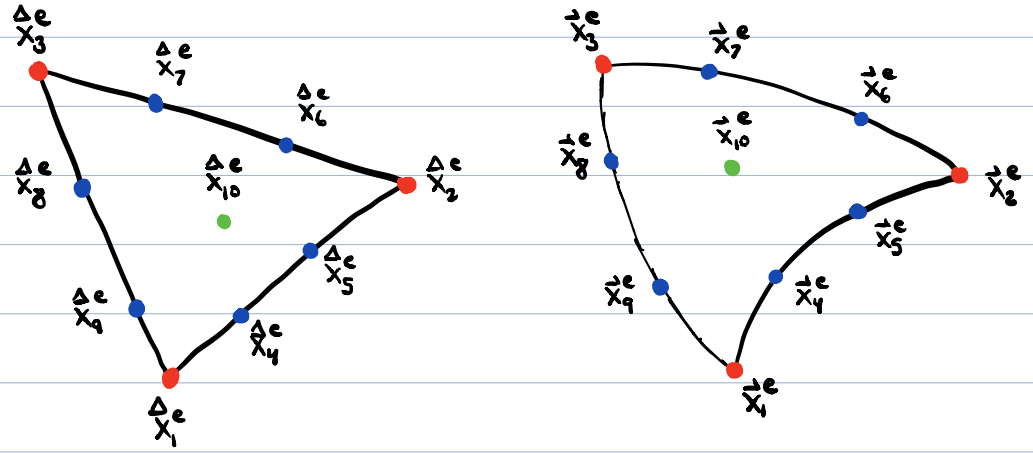
$$\hat{x}_a^e = \hat{x}_A$$

$$\vec{x}_a^e = \vec{x}_A$$

$$A = IEN(a,e)$$

In fact, since the Jacobian determinant of the mapping is positive, the orientation of the nodes is preserved. Visually:

$k=3$



Since:

$$\hat{N}_A\left(\overset{\Delta}{x}^e(\vec{\xi})\right) = \begin{cases} \hat{N}_a(\vec{\xi}) & \text{if there is an } a \text{ such that } A = ID(a,e) \\ 0 & \text{otherwise} \end{cases}$$

over each element Ω^e, it follows that:

$$\vec{x}\left(\overset{\Delta}{x}^e(\vec{\xi})\right) = \sum_{a=1}^{n_{en}} \vec{x}^e_a \hat{N}^e_a(\vec{\xi})$$

for $e = 1, \dots, n_{el}$ and $\vec{\xi} \in \hat{\Omega}$. Thus each element Ω^e can be described using a polynomial mapping:

$$\vec{x}^e(\vec{\xi}) = \sum_{a=1}^{n_{en}} \vec{x}^e_a \hat{N}^e_a(\vec{\xi})$$

from the parent element $\hat{\Omega}$ to the element Ω^e which depends only on the element

nodes $\{\vec{x}_a^e\}_{a=1}^{n_{en}}$ and not on the original triangular mesh $\hat{\mathcal{M}}^h$. Therefore, the mesh of curved elements, defined as:

$$\mathcal{M}^h := \{\Omega^e\}_{e=1}^{n_{el}}$$

can likewise be defined using solely the element connectivity, encoded in the IEN array, and the global nodes $\{x_A\}_{A=1}^{n_{nod}}$. Nonetheless, it is illustrative to think of the mesh of curved elements as the result of deforming a pre-existing triangular mesh.

Just as is the case for a triangular mesh, there is a domain associated with any mesh of curved elements:

$$\Omega^h := \text{int}\left\{\overline{\bigcup_{e=1}^{n_{el}} \Omega^e}\right\}$$

with boundary Γ^h. likewise, every curved boundary edge can be collected into a boundary mesh:

$$\mathcal{E}_{\Gamma}^h := \{\Gamma^e\}_{e=1}^{n_{elb}}$$

such that:

$$\Gamma^h = \overline{\bigcup_{e=1}^{n_{elb}} \Gamma^e}$$

Finally, a mesh of curved elements is suitable for finite element analysis if the following two conditions are satisfied:

Criterion 1: Each node lie in the domain of interest or its boundary.

Criterion 2: All nodes within or on the ends of a boundary edge lie either in the closure of the Dirichlet boundary, the closure of the Neumann boundary, or the closure of the Robin boundary.

If the above conditions apply, then we can define Dirichlet, Neumann, and Robin boundary meshes:

$$\mathcal{E}_D^h := \left\{ \Gamma^e \in \mathcal{E}_\Gamma^h : \vec{x}_A \in \overline{\Gamma_D^1} \text{ for all } \vec{x}_A \in \overline{\Gamma^e} \right\}$$

$$\mathcal{E}_N^h := \left\{ \Gamma^e \in \mathcal{E}_\Gamma^h : \vec{x}_A \in \overline{\Gamma_N^1} \text{ for all } \vec{x}_A \in \overline{\Gamma^e} \right\}$$

$$\mathcal{E}_R^h := \left\{ \Gamma^e \in \mathcal{E}_\Gamma^h : \vec{x}_A \in \overline{\Gamma_R^1} \text{ for all } \vec{x}_A \in \overline{\Gamma^e} \right\}$$

such that:

$$\mathcal{E}_\Gamma^h = \mathcal{E}_D^h \cup \mathcal{E}_N^h \cup \mathcal{E}_R^h \quad \text{and} \quad \mathcal{E}_D^h \cap \mathcal{E}_N^h = \mathcal{E}_N^h \cap \mathcal{E}_R^h = \mathcal{E}_R^h \cap \mathcal{E}_D^h = \emptyset$$

and Dirichlet, Neumann, and Robin finite element boundaries:

$$\Gamma_D^h := \text{int}\left(\overline{\bigcup_{\Gamma^e \in \mathcal{E}_D^h} \Gamma^e} \right)$$

$$\Gamma_N^h := \text{int}\left(\overline{\bigcup_{\Gamma^e \in \mathcal{E}_N^h} \Gamma^e} \right)$$

$$\Gamma_R^h := \text{int}\left(\overline{\bigcup_{\Gamma^e \in \mathcal{E}_R^h} \Gamma^e} \right)$$

such that:

$$\Gamma^h = \overline{\Gamma_D^h \cup \Gamma_N^h \cup \Gamma_R^h} \quad \text{and} \quad \Gamma_D^h \cap \Gamma_N^h = \Gamma_N^h \cap \Gamma_R^h = \Gamma_R^h \cap \Gamma_D^h = \emptyset$$

It should be noted that while meshes of curved elements are more accurate representations of the domain geometry of interested than meshes of straight-sided elements, they are generally still inexact. So, in general:

$$\Omega^h \approx \Omega \qquad \Gamma^h \approx \Gamma$$

$$\Gamma_D^h \approx \Gamma_D \qquad \Gamma_N^h \approx \Gamma_N \qquad \Gamma_R^h \approx \Gamma_R$$

However, the geometry approximation associated with a mesh of curved elements generally pollutes the accuracy of a finite element approximation less than that associated with a mesh of straight-sided elements. In particular, isoparametric finite element approximations of steady two-dimensional heat conduction exhibit optimal convergence rates in the H^1-norm regardless of polynomial degree.

The task of generating a mesh of curved elements, referred to as high-order mesh generation, is much more challenging than the task of generating a mesh of straight-sided elements. Modern algorithms for generating meshes of curved triangular elements are generally composed of four steps:

Step 1: Generation of a mesh of straight-sided triangular elements.

Step 2: Placement of vertex, edge, and element nodes.

Step 3: Motion of vertex and edge nodes along the finite element boundary to the boundary of the domain of interest.

Step 4: Motion of vertex, edge, and element nodes within the finite element domain to arrive at a high-quality mesh.

The first three steps above are straight-forward. The last step, however, is much more challenging. Ideally, the nodes of the mesh should be placed so that the Jacobian determinants of the element mappings:

$$\det \begin{bmatrix} \frac{\partial x_1^e}{\partial \xi_1} & \frac{\partial x_1^e}{\partial \xi_2} \\ \frac{\partial x_2^e}{\partial \xi_1} & \frac{\partial x_2^e}{\partial \xi_2} \end{bmatrix}$$

not only be positive but also as large as possible. This leads to a very nonlinear problem. In practice, linear mesh smoothing algorithms based on harmonic maps, biharmonic maps, or linear elasticity are instead employed to optimize the location of interior nodes. While these algorithms are not guaranteed to generate high-quality meshes, they typically do in practice and are much more efficient than nonlinear mesh smoothing algorithms with quality guarantees.

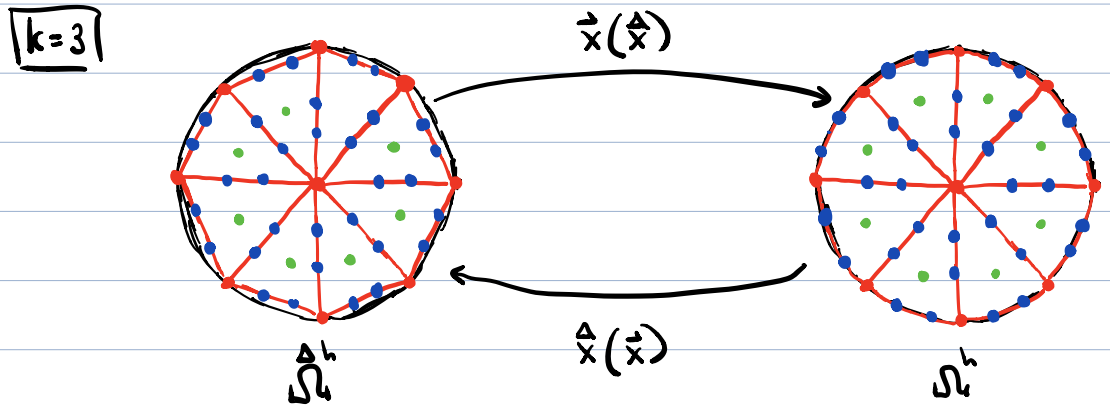
While it is straight-forward to construct spaces of continuous piecewise polynomial functions on a mesh of straight-sided triangular elements, the same is not true of meshes of curved triangular elements. As such, we will turn to so-called isoparametric finite element functions here. These rely on the notion of pull back and push forward. The pull back of a function $v : \Omega_h^h \to \mathbb{R}$ defined over the curved finite element domain to the triangular finite element domain is defined as:

$$(v \circ \vec{x})(\hat{\vec{x}}) = v(\vec{x}(\hat{\vec{x}}))$$

for all $\hat{\vec{x}} \in \hat{\Omega}_h^h$. Similarly, the push forward of a function $\hat{v} : \hat{\Omega}_h \to \mathbb{R}$ defined over the triangular finite element domain to the curved finite element domain is defined as:

$$(\hat{v} \circ \hat{\vec{x}})(\vec{x}) = \hat{v}(\hat{\vec{x}}(\vec{x}))$$

for all $\vec{x} \in \Omega_h^h$ where $\hat{\vec{x}} : \Omega_h^h \to \mathbb{R}^2$ is the inverse of $\vec{x} : \hat{\Omega}_h^h \to \mathbb{R}^2$:



Then an isoparametric finite element function is a function defined over the curved finite element domain whose pull back to the triangular finite element domain is a continuous piecewise polynomial of degree k with respect to the triangular finite element mesh. The isoparametric finite element approximation space is then:

$$\mathcal{P}^k_{\text{Cont}}\left(\mathcal{M}^h, \{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}\right) = \left\{ v^h : \Omega_h^h \to \mathbb{R} : \underbrace{v^h \circ \vec{x}}_{\text{Pull back of } v^h \text{ to } \hat{\Omega}_h^h} \in \hat{\mathcal{P}}^k_{\text{Cont.}}(\hat{\mathcal{M}}^h) \right\}$$

or equivalently:

$$\mathcal{P}^k_{\text{cont}}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}\right) = \left\{ \underbrace{\overset{\Delta}{v}{}^h \circ \overset{\Delta}{\vec{x}}}_{\text{Push forward of } \overset{\Delta}{v}{}^h \text{ to } \Omega^h} : \overset{\Delta}{v}{}^h \in \mathcal{P}^k_{\text{cont.}}\left(\overset{\Delta}{\mathcal{M}}{}^h\right) \right\}$$

Note the notation makes it clear that $\mathcal{P}^k_{\text{cont}}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}\right)$ depends not only on the curved elements $\{\Omega^e\}_{e=1}^{n_{\text{el}}}$ but <u>also</u> the nodes $\{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}$. Note also that:

$$\mathcal{P}^k_{\text{cont}}\left(\overset{\Delta}{\mathcal{M}}{}^h\right) = \mathcal{P}^k_{\text{cont}}\left(\overset{\Delta}{\mathcal{M}}{}^h; \{\overset{\Delta}{\vec{x}}_A\}_{A=1}^{n_{\text{nod}}}\right)$$

So our notation for isoparametric triangular finite element approximation spaces is consistent with our earlier notation for affine triangular finite element approximation spaces. Note that, by construction, any isoparametric finite element function $v^h \in \mathcal{P}^k_{\text{cont}}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}\right)$ takes the form:

$$v^h(\vec{x}) = \sum_{A=1}^{n} \underbrace{(v^h \circ \vec{x})(\overset{\Delta}{\vec{x}}_A)}_{\text{Degree of Freedom}} \underbrace{\left(\overset{\Delta}{N}_A \circ \overset{\Delta}{\vec{x}}\right)(\vec{x})}_{\text{Basis Function}}$$

Noting $(v^h \circ \vec{x})(\overset{\Delta}{\vec{x}}_A) = v^h(\vec{x}_A)$ and defining:

$$N_A(\vec{x}) = \underbrace{\left(\overset{\Delta}{N}_A \circ \overset{\Delta}{\vec{x}}\right)}_{\text{Push forward of } \overset{\Delta}{N}_A \text{ to } \Omega^h}(\vec{x})$$

we see $\{N_A\}_{A=1}^{n_{\text{nod}}}$ is a <u>basis</u> for $\mathcal{P}^k_{\text{cont}}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{\text{nod}}}\right)$ with

corresponding degrees of freedom $\{v^h(x_A)\}_{A=1}^{n_{nod}}$. The basis functions are called Lagrange basis functions as they are the push forwards of Lagrange basis functions over the triangular finite element mesh, and the degrees of freedom are likewise called nodal degrees of freedom.

Note that:

$$v^h \circ \vec{x}^e \in P^k(\hat{\Omega})$$

for $e = 1, \ldots, n_{el}$ for any $v^h \in P^k_{cont}(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}})$. This inspires the following alternative definition of $P^k_{cont}(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}})$:

$$P^k_{cont}(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}}) := \left\{ v^h \in C^0(\bar{\Omega}^h) : \underbrace{v^h \circ \vec{x}^e}_{\text{Pull back of finite element function } v^h \text{ to parent element } \hat{\Omega}} \in P^k(\hat{\Omega}),\ e = 1, \ldots, n_{el} \right\}$$

In fact, the above definition is much more common in the finite element literature than the one presented earlier. We simply worked with a different definition as it was simpler to arrive at a basis using this definition. The meaning of the word isoparametric is clear when looking at the above definition: the degree of the finite element parameterization $\vec{x}^e$ is the same as that of the pull back of finite element function $v^h \circ \vec{x}^e$. In a subparametric finite element approximation, the degree of $\vec{x}^e$ is less than that of $v^h \circ \vec{x}^e$, while in a superparametric finite element approximation, the degree of $\vec{x}^e$ is greater than that of $v^h \circ \vec{x}^e$. In an affine finite element approximation, the degree of $\vec{x}^e$ is precisely one.

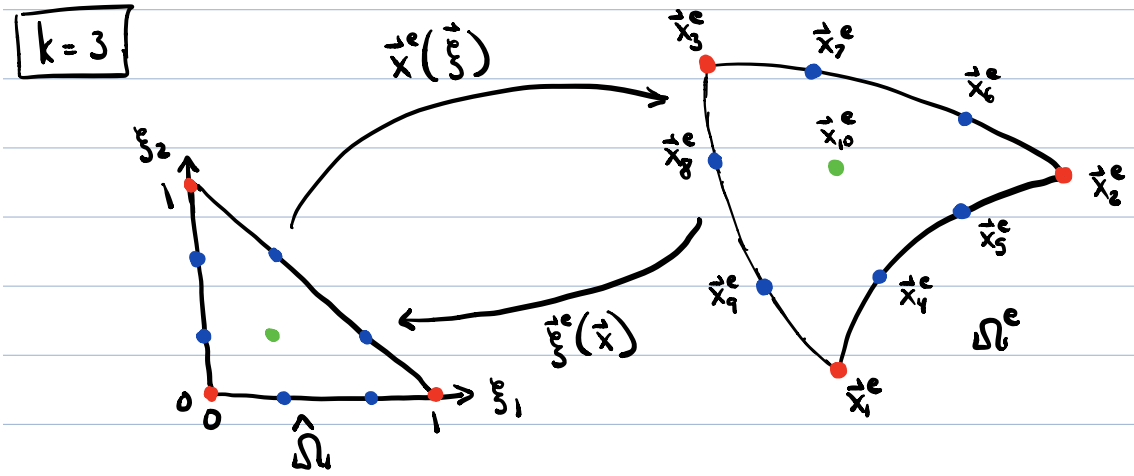
We have that:

$$(N_A \circ \vec{x}^e)(\vec{\xi}) = (\hat{N}_A \circ \overset{\Delta e}{x})(\vec{\xi}) = \begin{cases} \hat{N}_a(\vec{\xi}) & \text{if there is an } a \text{ such that } A = \mathrm{IEN}(a,e) \\ 0 & \text{otherwise} \end{cases}$$

for each element Ω^e. Thus we can express the basis functions $\{N_A\}_{A=1}^{n_{nod}}$ over each element Ω^e as push forwards of the shape functions $\{\hat{N}_a\}_{a=1}^{n_{en}}$ over the parent element:

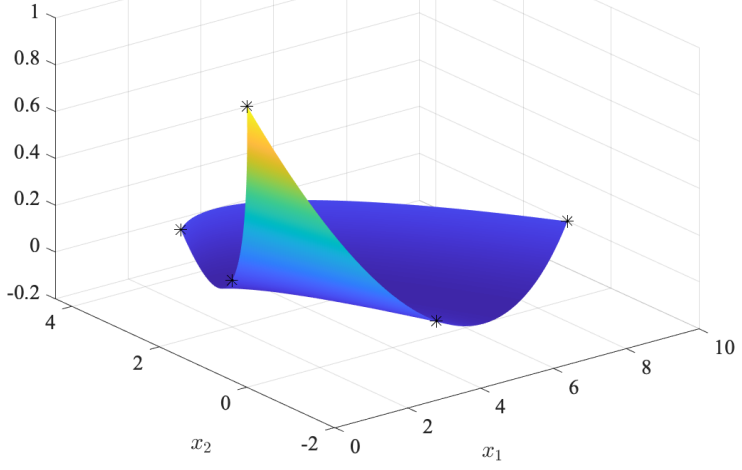
$$N_A(\vec{x}) = \begin{cases} (\hat{N}_a \circ \overset{e}{\xi})(\vec{x}) & \text{if there is an } a \text{ such that } A = \mathrm{IEN}(a,e) \\ 0 & \text{otherwise} \end{cases}$$

where $\vec{\xi}^e : \Omega^e \to \hat{\Omega}$ is the inverse of $\vec{x}^e : \hat{\Omega} \to \Omega^e$:

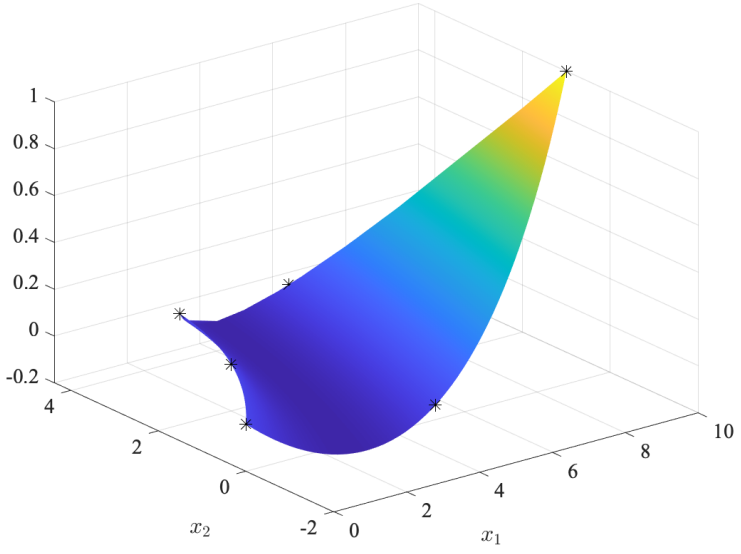


Plots of the push forwards of quadratic shape functions for a representative element mapping are included on the next page. We will exploit the above relation to later arrive at efficient element formation routines for isoparametric finite element approximations.

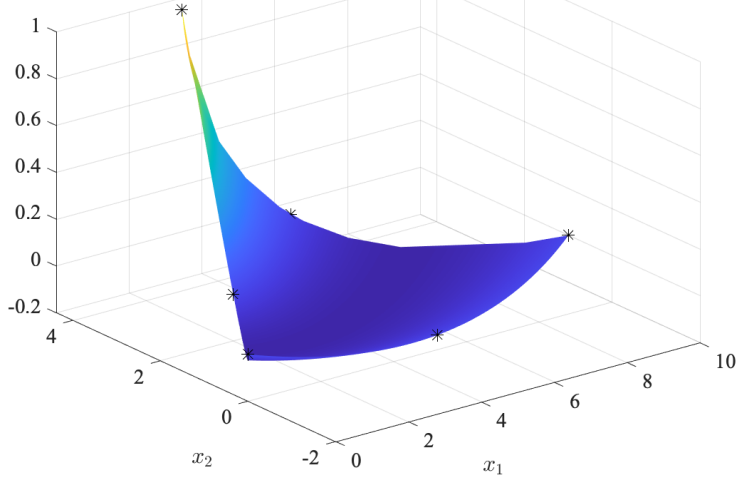
TRI $k = 2$: $N_1(x_1, x_2)$



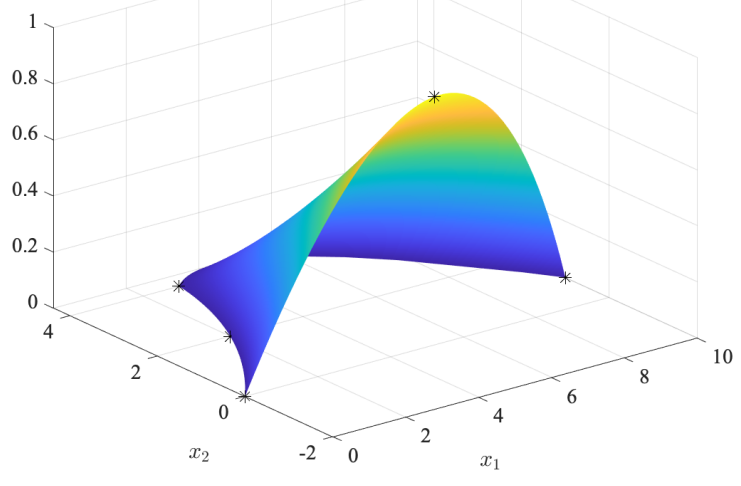
TRI $k = 2$: $N_2(x_1, x_2)$



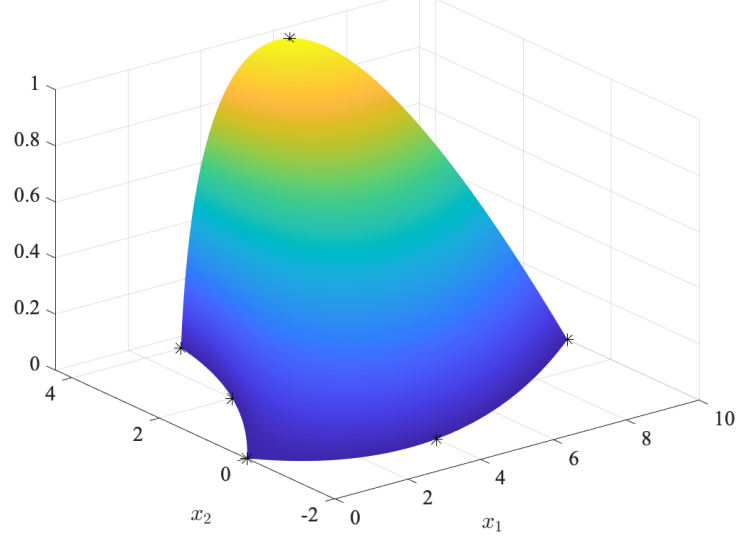
TRI $k = 2$: $N_3(x_1, x_2)$



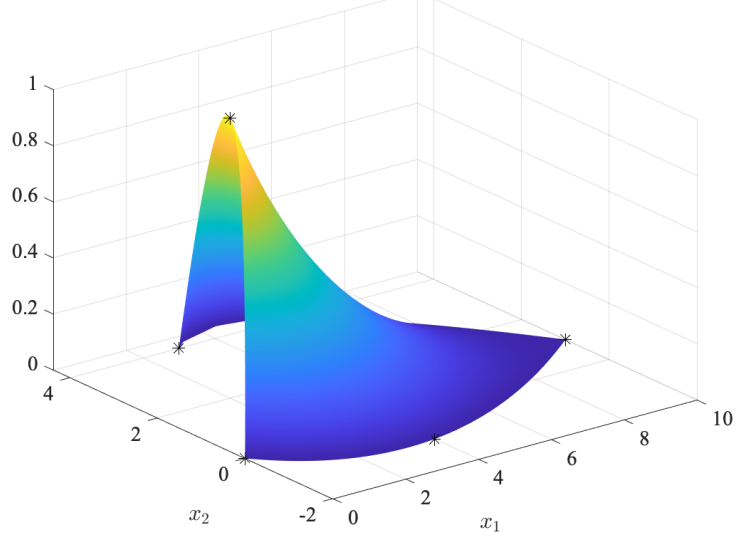
TRI $k = 2$: $N_4(x_1, x_2)$



TRI $k = 2$: $N_5(x_1, x_2)$



TRI $k = 2$: $N_6(x_1, x_2)$



The Lagrange basis functions $\{N_A\}_{A=1}^{n_{nod}}$ satisfy the same properties as Lagrange basis functions over a triangular finite element mesh:

1. They are interpolatory. That is:

$$N_A(\vec{x}_B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{if } A \neq B \end{cases}$$

2. They are locally supported. In particular:

$$N_A(\vec{x}) = 0 \quad \text{if } \vec{x} \in \bar{\Omega}_i^c \text{ and } \vec{x}_A \notin \bar{\Omega}_i^e$$

3. They form a partition of unity. That is:

$$\sum_{A=1}^{n_{nod}} N_A(\vec{x}) = 1 \quad \text{for } \vec{x} \in \Omega$$

4. They are non-negative for $k=1$. That is:

$$N_A(\vec{x}) \geq 0 \quad \text{for } \vec{x} \in \Omega$$

if $k=1$.

Moreover, as:

$$\vec{x} = \sum_{A=1}^{n_{nod}} \vec{x}_A N_A(\vec{x})$$

every linear polynomial belongs to $P^k_{cont}(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}})$. In particular,

the function $v^h(\vec{x}) = c_1 x_1 + c_2 x_2 + c_3$ admits the form:

$$v^h(\vec{x}) = \sum_{A=1}^{n_{nod}} \left(c_1 (\vec{x}_A)_1 + c_2 (\vec{x}_A)_2 + c_3 \right) N_A(\vec{x})$$

The property that $\mathcal{P}^k_{cont}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}}\right)$ contains linear polynomials is often called Completeness. Local smoothness, global continuity, and completeness are often cited as "requirements" for convergence of a finite element approximation (see, e.g., Section 3.1 of Hughes' text "The Finite Element Method: linear Static and Dynamic Finite Element Analysis). However, contrary to popular belief, these conditions are neither necessary nor sufficient for convergence. Nonetheless, completeness is often a desirable property for a finite element approximation space to have.

With an isoparametric finite element approximation space in hand, we define the corresponding set of finite element trial solutions as:

$$\mathcal{S}^h := \left\{ v^h \in \mathcal{P}^k_{cont}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}}\right) : v^h(\vec{x}_A) = g(\vec{x}_A) \text{ for } A \in \eta_D \right\}$$

and the corresponding space of finite element test functions as:

$$\mathcal{V}^h := \left\{ v^h \in \mathcal{P}^k_{cont}\left(\mathcal{M}^h; \{\vec{x}_A\}_{A=1}^{n_{nod}}\right) : v^h(\vec{x}_A) = 0 \text{ for } A \in \eta_D \right\}$$

where, as in the affine setting:

$$\eta_D := \left\{ A \in \eta : \vec{x}_A \in \overline{\Gamma_D^h} \right\} \quad \text{and} \quad \eta := \{1, \ldots, n_{nod}\}$$

Then, an isoparametric triangular finite element approximation of two-dimensional steady heat conduction using the above approximations takes the form:

(G) Find $T^h \in \mathcal{S}^h$ such that:

$$b^h(T^h, w^h) = l^h(w^h) \qquad \text{for all } w^h \in \mathcal{V}^h$$

where:

$$b^h(T^h, w^h) = \int_{\Omega_h^h} K \vec{\nabla} T^h \cdot \vec{\nabla} w^h \, d\Omega_h + \int_{\Gamma_R^h} \beta T^h w^h \, d\Gamma^2$$

$$l^h(w^h) = \int_{\Omega_h^h} f w^h \, d\Omega_h + \int_{\Gamma_N^h} h w^h \, d\Gamma^2 + \int_{\Gamma_R^h} \beta T_R w^h \, d\Gamma^2$$

and, just as in the affine setting, the degrees of freedom of the Galerkin solution:

$$T^h(x) = \sum_{A \in \eta - \eta_D} T^h(\vec{x}_A) N_A(\vec{x}) + \sum_{A \in \eta_D} T^h(\vec{x}_A) N_A(\vec{x})$$

$T^h(\vec{x}_A)$: Degree of Freedom

$T^h(\vec{x}_A) = g(\vec{x}_A)$

$\sum_{A \in \eta_D} T^h(\vec{x}_A) N_A(\vec{x}) = g^h(\vec{x})$

may be attained via the solution of the linear system:

$$(L)\begin{cases} \text{Find } \underline{d} \in \mathbb{R}^{n_{eq}} \text{ such that:} \\ \underline{\underline{K}}\,\underline{d} = \underline{F} \end{cases}$$

where:

$$K_{PQ} = b^h(N_B, N_A) \qquad \text{where} \quad ID(A) = P,\ ID(B) = Q$$

$$F_P = l^h(N_A) - b^h(g^h, N_A) \qquad \text{where} \quad ID(A) = P$$

$$d_Q = T^h(\vec{x}_B) \qquad \text{where} \quad ID(B) = Q$$

and ID is a destination array assigning each degree of freedom index $A \in \eta - \eta_D$ a unique equation number between 1 and the total number of degrees of freedom n_{eq} and assigning each Dirichlet index $A \in \eta_D$ a value of 0.