Isoparametric Finite Element Mappings

Corbin Foucart

1 Isoparametric Mappings

1.1 mapping from reference element to physical space

We express the transformation from reference space to physical space as a linear combination of the basis functions of the chosen finite element space. The concise statement can be written:

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{j=1}^{n_b} \boldsymbol{x}_j^K \psi_j(\boldsymbol{\xi}) \tag{1}$$

Where n_b is the number of nodal shape functions defined on the master element, and $\boldsymbol{x}_j^K$ are the interpolation points on the physical space triangle. Note that the number of interpolation points in physical space must be the same as the number of basis functions. Therefore the 'interpolation points' are the nodal points in physical space. We can clarify the above expression by writing the explicit statement in two dimensions:

$$x(\xi, \eta) = \sum_{j=1}^{n_b} x_j^K \psi_j(\xi, \eta) \tag{2}$$

$$y(\xi, \eta) = \sum_{j=1}^{n_b} y_j^K \psi_j(\xi, \eta) \tag{3}$$

So if we would like to map any set of points (here we choose quadrature points $\boldsymbol{\xi}_q$, but the approach is general and can be used for any set of points $\boldsymbol{\xi}_\alpha$ defined on the master element) defined on the master element to their corresponding points in physical space, we can use the mapping directly for each quadrature point $\boldsymbol{\xi}_q$:

$$\boldsymbol{x}(\boldsymbol{\xi}_q) = \sum_{j=1}^{n_b} \boldsymbol{x}_j^K \psi_j(\boldsymbol{\xi}_q) \qquad \rightarrow \qquad \underbrace{\begin{bmatrix} \psi_1(\boldsymbol{\xi}_1), & \cdots, & \psi_{n_b}(\boldsymbol{\xi}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{\xi}_q), & \cdots, & \psi_{n_b}(\boldsymbol{\xi}_q) \end{bmatrix}}_{G} \begin{bmatrix} x_1^K, & y_1^K \\ \vdots & \vdots \\ x_{n_b}^K, & y_{n_b}^K \end{bmatrix} \tag{4}$$

In order to map any set of points on the master element $\boldsymbol{\xi}_\alpha$ (indexed here by α), we need the values of the nodal shape functions evaluated at those points $\boldsymbol{\xi}_\alpha$ and the physical space nodal points $\boldsymbol{x}_j^K$. Note that in the case of mapping the nodal points on the master element to the physical space nodes, note that the transformation matrix G is simply the identity matrix I (nodal basis), and the physical space nodal points are recovered.

1.2 computation of mapping derivatives, jacobian J

In this subsection, we use the explicit example of the derivatives of the isoparametric transformation in 2D, for clarity. The key idea is that since we have an explicit representation of our transformation in (1), we can take the derivative of the transformation with respect to master element coordinates ξ, η

$$\frac{\partial x}{\partial \xi} = \frac{\partial x(\xi, \eta)}{\partial \xi} = \frac{\partial}{\partial \xi}\left[\sum_{j=1}^{n_b} x_j^K \psi_j(\xi, \eta)\right] = \sum_{j=1}^{n_b} x_j^K \frac{\partial \psi_j}{\partial \xi} \tag{5}$$

The last expression of (5) is easily computable since we know both the derivates of the shape functions ψ_j on the master element with respect to the master coordinates ξ, η. The Jacobian matrix of the transformation from the reference element $\widehat{K}$ to the physical space element K at a specific point $\boldsymbol{x}_\alpha$ with corresponding point $\boldsymbol{\xi}_\alpha$ on the master element can be written

$$J\big|_{\boldsymbol{x}_\alpha} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}_{\boldsymbol{x}_\alpha} = \begin{bmatrix} \sum_{j=1}^{n_b} x_j^K \frac{\partial \psi_j(\boldsymbol{\xi}_\alpha)}{\partial \xi} & \sum_{j=1}^{n_b} y_j^K \frac{\partial \psi_j(\boldsymbol{\xi}_\alpha)}{\partial \xi} \\ \sum_{j=1}^{n_b} x_j^K \frac{\partial \psi_j(\boldsymbol{\xi}_\alpha)}{\partial \eta} & \sum_{j=1}^{n_b} y_j^K \frac{\partial \psi_j(\boldsymbol{\xi}_\alpha)}{\partial \eta} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{\partial \psi_1(\boldsymbol{\xi}_\alpha)}{\partial \xi} & \cdots & \frac{\partial \psi_{n_b}(\boldsymbol{\xi}_\alpha)}{\partial \xi} \\ \frac{\partial \psi_1(\boldsymbol{\xi}_\alpha)}{\partial \eta} & \cdots & \frac{\partial \psi_{n_b}(\boldsymbol{\xi}_\alpha)}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1^K, & y_1^K \\ \vdots & \vdots \\ x_{n_b}^K, & y_{n_b}^K \end{bmatrix} \tag{6}$$

Note that (6) is directly computable from the derivatives of the shape functions, evaluated at the point of interest $\boldsymbol{\xi}_\alpha$ on the master element, as well as the physical nodal points. In practice, we evaluate the derivatives of the nodal shape functions at the quadrature points and nodal points (for quadrature-based and quadrature-free integrals, respectively). We will denote $\det(J)$ as $|J|$. For every point pairing $\boldsymbol{\xi}_\alpha$, $\boldsymbol{x}_\alpha$ on the reference element and physical space (respectively), once we have computed $J(\boldsymbol{x}_\alpha)$, computing $|J|$ at $\boldsymbol{x}_\alpha$ is simply a matter of computing the determinant.

1.3   computation of inverse mapping derivatives, J^{-1}

We will explicitly need to compute the derivates of the inverse mapping from physical space to reference space (discussed in 1.4) which arises from the need to compute terms of the form $\frac{\partial \phi_i}{\partial x}$, i.e., the derivatives of the global nodal basis functions with respect to physical space coordinates. The difficulty is that computing and storing the derivatives of all global basis functions at every point of interest in the computational domain would be enormously expensive. Instead, we wish to map the derivatives of the global basis functions to the derivatives of the shape functions on the master element, e.g., $\frac{\partial \psi_i(\boldsymbol{\xi})}{\partial \xi}$, which are known and stored. We can link the two via the chain rule. In one spatial dimension, (1) can be applied with the chain rule to conclude

$$\frac{\partial \phi_i}{\partial \xi} = \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \xi} \tag{7}$$

Since we have the functional dependence $\phi_i(x(\xi))$. In two dimensions, we have the functional dependence
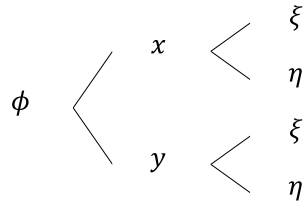
Hence we can write the chain rule statement applied over the isoparametric mapping as

$$\frac{\partial \phi_i}{\partial \xi} = \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial \phi_i}{\partial y} \frac{\partial y}{\partial \xi}, \qquad \frac{\partial \phi_i}{\partial \eta} = \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial \phi_i}{\partial y} \frac{\partial y}{\partial \eta} \tag{8}$$

$$\Rightarrow \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \end{bmatrix} = J \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \end{bmatrix} \tag{9}$$

Where J is the jacobian matrix computed at a physical space point $\boldsymbol{x}_\alpha$ in the previous section. We can directly invert this expression to find the inverse transform J^{-1}, which will allow us to compute the derivatives of the global basis functions in terms of the derivatives of the master element shape functions.

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \end{bmatrix} \tag{10}$$

Which can be explicitly written as

$$\begin{aligned} \frac{\partial \phi_i}{\partial x} &= \frac{\partial \phi_i}{\partial \xi} J_{11}^{-1} + \frac{\partial \phi_i}{\partial \eta} J_{12}^{-1} \\ \frac{\partial \phi_i}{\partial y} &= \frac{\partial \phi_i}{\partial \xi} J_{21}^{-1} + \frac{\partial \phi_i}{\partial \eta} J_{22}^{-1} \end{aligned} \tag{11}$$

Extension to the general case should be apparent from the specific two-dimensional examples.

1.4 continuous integral operators

We can write the forms of the continuous integral operators by directly substituting the inverted chain rule relations (11) into the physical space integral expressions.

1.4.1 interior operators

Weak laplacian:

$$\begin{aligned} \int_K \nabla \phi_j \cdot \nabla \phi_i \, d\boldsymbol{x} &= \int_K \left(\frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_i}{\partial y} \frac{\partial \phi_j}{\partial y} \right) d\boldsymbol{x} \\ &= \int_{\widehat{K}} \left(\frac{\partial \widehat{\phi_i}}{\partial \xi} J_{11}^{-1} + \frac{\partial \widehat{\phi_i}}{\partial \eta} J_{12}^{-1} \right) \left(\frac{\partial \widehat{\phi_j}}{\partial \xi} J_{11}^{-1} + \frac{\partial \widehat{\phi_j}}{\partial \eta} J_{12}^{-1} \right) |J| \, d\widehat{\boldsymbol{x}} \\ &\quad + \int_{\widehat{K}} \left(\frac{\partial \widehat{\phi_i}}{\partial \xi} J_{21}^{-1} + \frac{\partial \widehat{\phi_i}}{\partial \eta} J_{22}^{-1} \right) \left(\frac{\partial \widehat{\phi_j}}{\partial \xi} J_{21}^{-1} + \frac{\partial \widehat{\phi_j}}{\partial \eta} J_{22}^{-1} \right) |J| \, d\widehat{\boldsymbol{x}} \end{aligned} \tag{12}$$

Convection-like derivatives:

$$\begin{aligned} \int_K \nabla \phi_i \cdot \boldsymbol{\phi}_j \, d\boldsymbol{x} &= \int_K \left(\frac{\partial \phi_i}{\partial x} \phi_j + \frac{\partial \phi_i}{\partial y} \phi_j \right) d\boldsymbol{x} \\ &= \int_{\widehat{K}} \left(\frac{\partial \widehat{\phi_i}}{\partial \xi} J_{11}^{-1} + \frac{\partial \widehat{\phi_i}}{\partial \eta} J_{12}^{-1} \right) \phi_j |J| \, d\widehat{\boldsymbol{x}} + \int_{\widehat{K}} \left(\frac{\partial \widehat{\phi_i}}{\partial \xi} J_{21}^{-1} + \frac{\partial \widehat{\phi_i}}{\partial \eta} J_{22}^{-1} \right) \phi_j |J| \, d\widehat{\boldsymbol{x}} \end{aligned} \tag{13}$$

Source term integration:

$$\int_K f_\Omega \phi_i \, d\boldsymbol{x} = \int_{\widehat{K}} \widehat{f}_\Omega \, \widehat{\phi}_i |J| \, d\widehat{\boldsymbol{x}} = \int_{\widehat{K}} f_\Omega(\boldsymbol{\xi}(\boldsymbol{x})) \widehat{\phi}_i |J| \, d\widehat{\boldsymbol{x}} \tag{14}$$

Where there is some subtlety in $\widehat{f}_\Omega$, since we are integrating the $f_\Omega(\boldsymbol{x})$ over some K mapped to the reference element $\widehat{K}$. This inverse mapping $\boldsymbol{\xi}(\boldsymbol{x})$ maps physical space locations to their corresponding locations on the master element. While in general, this inverse mapping is equivalent to solving a non-linear set of equations, in practice, we don't need to compute the inverse map, because the points at which we evaluate the function in physical space (either nodal points or quadrature points) are mapped from the master element nodal or quadrature points – i.e., the points on the master element which correspond to the physical space nodal/quadrature points are simply the nodal/quadrature points on the master element, by construction!

1.5 discrete integral operators

1.5.1 quadrature-based operators

We can choose a classical quadrature scheme (points and weights defined on the master element) and approximate the continuous form of the weak laplacian operator (12) accordingly. We done the master element quadrature points by $\boldsymbol{\xi}_q$ and corresponding mapped physical space quadrature points $\boldsymbol{x}_q$.

Weak laplacian:

$$
\int_K \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x} \approx \sum_{q=1}^{n_q} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q)J_{11}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q)J_{12}^{-1}(\boldsymbol{x}_q) \right) \left(\frac{\partial\widehat{\phi_j}}{\partial\xi}(\boldsymbol{\xi}_q)J_{11}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_j}}{\partial\eta}(\boldsymbol{\xi}_q)J_{12}^{-1}(\boldsymbol{x}_q) \right) w_q |J|\Big|_{\boldsymbol{x}_q}
$$
$$
+ \sum_{q=1}^{n_q} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q)J_{21}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q)J_{22}^{-1}(\boldsymbol{x}_q) \right) \left(\frac{\partial\widehat{\phi_j}}{\partial\xi}(\boldsymbol{\xi}_q)J_{21}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_j}}{\partial\eta}(\boldsymbol{\xi}_q)J_{22}^{-1}(\boldsymbol{x}_q) \right) w_q |J|\Big|_{\boldsymbol{x}_q}
\tag{15}
$$

Convection-like:

$$
\int_K \nabla\phi_i \cdot \boldsymbol{\phi}_j \, d\boldsymbol{x} \approx \sum_{q=1}^{n_q} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q)J_{11}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q)J_{12}^{-1}(\boldsymbol{x}_q) \right) \phi_j(\boldsymbol{\xi}_q)w_q |J|\Big|_{\boldsymbol{x}_q}
$$
$$
+ \sum_{q=1}^{n_q} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q)J_{21}^{-1}(\boldsymbol{x}_q) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q)J_{22}^{-1}(\boldsymbol{x}_q) \right) \phi_j(\boldsymbol{\xi}_q)w_q |J|\Big|_{\boldsymbol{x}_q}
\tag{16}
$$

1.5.2 quadrature-free operators

It is possible to form integral operators without using a "traditional" quadrature scheme (points and weights chosen on the master element distinct from the nodal points), such as described in [2]. In the discussion above, we were careful to describe the isoparametric mapping at arbitrary spatial locations. Instead of computing the det J and J^{-1} quantities at the quadrature points in physical space, we compute the data at the nodal points only.

Because the scheme in [2] is used in a mixed-method finite element formulation (HDG) using the so-called "strong form" DG-FEM, the weak laplacian as written above does not appear. Instead, the integral derivative operators are similar to the convection-like operators in (13).

Convection-like:

$$
\int_K \nabla\phi_i \cdot \boldsymbol{\phi}_j \, d\boldsymbol{x} \approx \sum_{q=1}^{n_b} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{11}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{12}^{-1}(\boldsymbol{x}_q^n) \right) \phi_j(\boldsymbol{\xi}_q^n)|J|\Big|_{\boldsymbol{x}_q^n}
$$
$$
+ \sum_{q=1}^{n_b} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{21}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{22}^{-1}(\boldsymbol{x}_q^n) \right) \phi_j(\boldsymbol{\xi}_q^n)|J|\Big|_{\boldsymbol{x}_q^n}
\tag{17}
$$

Where we emphasize that there are no weights and all quantities of interest are evaluated at nodal points $\boldsymbol{\xi}_q^n$ or $\boldsymbol{x}_q^n$ rather than quadrature points. This works acceptably for straight-sided elements, but does it work for curved elements? by analogy, we create the quadrature free weak laplacian operator.

Weak laplacian (analogy):

$$
\int_K \nabla\phi_j \cdot \nabla\phi_i \, d\boldsymbol{x} \approx \sum_{q=1}^{n_b} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{11}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{12}^{-1}(\boldsymbol{x}_q^n) \right) \left(\frac{\partial\widehat{\phi_j}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{11}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_j}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{12}^{-1}(\boldsymbol{x}_q^n) \right) |J|\Big|_{\boldsymbol{x}_q^n}
$$
$$
+ \sum_{q=1}^{n_b} \left(\frac{\partial\widehat{\phi_i}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{21}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_i}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{22}^{-1}(\boldsymbol{x}_q^n) \right) \left(\frac{\partial\widehat{\phi_j}}{\partial\xi}(\boldsymbol{\xi}_q^n)J_{21}^{-1}(\boldsymbol{x}_q^n) + \frac{\partial\widehat{\phi_j}}{\partial\eta}(\boldsymbol{\xi}_q^n)J_{22}^{-1}(\boldsymbol{x}_q^n) \right) |J|\Big|_{\boldsymbol{x}_q^n}
\tag{18}
$$

References

[1] Nico Schloemer. `quadpy`: numerical integration in python, 2018.

[2] M. P. Ueckermann and P. F. J. Lermusiaux. Hybridizable discontinuous Galerkin projection methods for Navier–Stokes and Boussinesq equations. *Journal of Computational Physics*, 306:390–421, 2016.