Introduction to Spectral Element Methods

BY HAIJUN YU

Email: yuhaijun@gmail.com

1 One-Dimensional Spectral Element Method

We use the 1-dimensional elliptic Helmholtz equation as a model equation.

$$\begin{cases} \alpha u - u'' = f, \quad x \in I = (0, L), \\ u(0) = u(L) = 0. \end{cases} \tag{1}$$

The corresponding weak form is

$$\begin{cases} \text{Find } u \in H_0^1(I), \quad \text{such that} \\ \alpha\,(u, v) + (u', v') = (f, v), \quad \forall v \in H_0^1(I). \end{cases} \tag{2}$$

1.1 Domain partition and Galerkin approximation

Partition to the interval I:

$$0 = x_0 < x_1 < \cdots < x_n = L; \qquad I = \bigcup_{i=1}^{n} I_i, \quad \text{with} \quad I_i = [x_{i-1}, x_i].$$

Approximation space

$$X^{n,p} = \{u(x) \in C^0(I),\ u|_{I_i} \in P_p\}, \quad X_0^{n,p} = \{\, u \in X^{n,p},\ u(0) = u(L) = 0\,\}, \tag{3}$$

The corresponding Garlerkin formalution

$$\begin{cases} \text{Find } u_N \in X_0^{n,p} \quad \text{such that} \\ \alpha\,(u_N, v) + (u_N', v') = (f, v), \quad \forall v \in X_0^{n,p}. \end{cases} \tag{4}$$

Map every interval $I_i = [x_{i-1}, x_i]$ to the standard interval $[-1, 1]$ by

$$x = m_i(\xi) = x_{i-1}\varphi_0(\xi) + x_i\varphi_1(\xi), \tag{5}$$

where

$$\varphi_0(\xi) = \frac{1-\xi}{2}, \quad \varphi_1(\xi) = \frac{1+\xi}{2}. \tag{6}$$

The Jocobi of this mapping is

$$J_i = \frac{\partial x}{\partial \xi} = \frac{x_i - x_{i-1}}{2} = \frac{|I_i|}{2} \tag{7}$$

Boundary-inner decomposition basis:

$$\{\,\phi_j(\xi)\,\}_{j=0}^{p} = \{\varphi_0(\xi), \varphi_1(\xi)\} \cup \{\,\phi_j(\xi),\ \phi_j(\pm 1) = 0\,\}_{j=2}^{p} \tag{8}$$

The above bases are in local variable. We extend them into global variable by

$$\psi_j^i(x) = \begin{cases} \phi_j(\xi), & \text{if } x \in I_i, \quad \xi = m_i^{-1}(x), \\ 0, & \text{if } x \notin I_i. \end{cases} \tag{9}$$

Note that for each $i = 1, ..., n-1$ that $\psi_1^i(x)$ and $\psi_0^{i+1}(x)$ are the two parts of the linear finite element hat function defined according to the three nodes $(x_{i-1},\ x_i,\ x_{i+1})$. By the continuous condition, the local degree of freedom corresponds to $\psi_1^i(x)$, and $\psi_0^{i+1}(x)$ refer to the same global degree of freedom. So the global basis set are

$$V = \{\psi_j(x)\}_{j=1}^{np-1} := \{\,\psi_1^i(x) + \psi_0^{i+1}(x)\,\}_{i=1}^{n-1} \cup \{\psi_j^i(x), j = 2, ..., p\}_{i=1}^{n} \tag{10}$$

In equation (4), we take $u_N = \tilde{u}_k\psi_k(x)$, $v = \psi_l(x)$, for all $l = 1, ..., np-1$, we get

$$(\alpha M + S)\boldsymbol{u} = \boldsymbol{f}, \tag{11}$$

where M and S are global mass and stiff matrices, their components are given by

$$M(l,k) = (\psi_k, \psi_l), \quad S(l,k) = (\psi_k', \psi_l').$$

1.2 Matrices Assembling

From

$$M(l,k) = (\psi_k, \psi_l) = \int_I \psi_k\psi_l \,\mathrm{d}x = \sum_{i=1}^{n} \int_{I_i} \psi_k\psi_l \,\mathrm{d}x,$$

we see that the global matrix can be calculated by take summation of nonzero local matrices in each element.

$$M_{j,j'}^{i,i'} := \int_I \psi_j^i\psi_{j'}^{i'}\mathrm{d}x = \delta_{ii'}\int_{I_i} \psi_j^i\psi_{j'}^i\mathrm{d}x = \delta_{ii'}\int_{-1}^{1} \phi_j(\xi)\phi_{j'}(\xi)J_i\mathrm{d}\xi = \delta_{ii'}J_i\, m_{jj'}$$

and

$$S_{j,j'}^{i,i'} := \int_I \psi_j^{i\prime}\psi_{j'}^{i\prime\prime}\,\mathrm{d}x = \delta_{ii'}\int_{I_i} \frac{\mathrm{d}\psi_j^i}{\mathrm{d}x}\frac{\mathrm{d}\psi_{j'}^{i'}}{\mathrm{d}x}\,\mathrm{d}x = \delta_{ii'}\int_{-1}^{1} \frac{\mathrm{d}\phi_j}{\mathrm{d}\xi}\frac{\mathrm{d}\phi_{j'}}{\mathrm{d}\xi}\frac{1}{J_i}\mathrm{d}\xi = \delta_{ii'}s_{jj'}/J_i$$

where $M_{j,j'}^{i,i'}$ and $S_{j,j'}^{i,i'}$ are the components of the unit mass and stiff matrices on element I_i, while $m_{jj'}$ and $s_{jj'}$ are the components of the mass and stiff matrices on the standard reference element. Because some global bases consist parts in different elements, which means different local DoF may link to the same global DoF, for those kind of global DoF, we need sum up the contributions from each elements. This procedure is called assembling. In the numerical implementation, one usually uses a mapping link local basis DoF index to global DoF index. Namely, $k = \mathrm{map}(i,\ j)$ stands for that the global DoF (Degree of Freedom) index of local j-th DoF in elemnet i is k. Following algorithm describes the procedure to build the linear system (12) by assembling.

Algorithm ASM (Assembling Procedure)
Input: $\{x_i\}_{i=0}^n$, $\{p_i \geqslant 1\}_{i=1}^n$; matrices $\{m_{jj'}\}_{j,j'=0}^p$ and $\{s_{jj'}\}_{j,j'=0}^p$, where $p = \max_i\{p_i\}$.
Output: Global matrices M and S

Step 1) Initialize the local to global index mapping:
for i from 1 to n
  map$(i,0)$=$i-1$; map$(i,1)$=i.
end for
map(1,0):=-1; map(n,1):=-1; //boundary points, not a DoF.
$k = n$
for i from 1 to n
  for j from 2 to p_i
    map(i,j)=k; $k = k+1$;
  end for
end for
Step 2) Assembling
Set $M = 0$ and $S = 0$.
for $i = 1$ to n
  for all the nonzero components m_{jl} in $\{m_{jl}\}_{j,l=0}^{p_i}$
    If map$(i,j) > 0$ and map$(i,j) > 0$ then $M(\mathrm{map}(i,j), \mathrm{map}(i,l)) \mathrel{+}= m_{jl} \times (x_i - x_{i-1})/2$
  end for
  for all the nonzero components s_{jl} in $\{s_{jl}\}_{j,l=0}^{p_i}$
    If map$(i,j) > 0$ and map$(i,j) > 0$ then $S(\mathrm{map}(i,j), \mathrm{map}(i,l)) \mathrel{+}= s_{jl} \times 2/(x_i - x_{i-1})$

end for
end for
End of the Algorithm

1.3 Choice of local basis

There are several choices of local spectral bases that results in sparse matrices.

1. For Chebyshev weighted Galerkin approximation, one may choose

$$\phi_j(\xi) = T_j(\xi) - T_{j-2}(\xi), \quad j \geqslant 2. \tag{12}$$

with T_j is the Chebyshev polynomial. In this case, Chebyshev weight function $(1 - \xi^2)^{-1/2}$ should be added to weak Galerkin formulation. This gives a tri-diagonal mass matrix and lower-triangular stiff matrix, with the resulting linear system can be solved in linear time. Another choice of Chebyshev basis is

$$\phi_j(\xi) = (1-\xi^2)T_{j-2}(\xi), \quad \text{for } j \geqslant 2.$$

2. For Legendre Galerkin approximation, One can use

$$\phi_j(\xi) = L_j(\xi) - L_{j-2}(\xi), \quad j \geqslant 2. \tag{13}$$

with L_j is the Legendre polynomial. This leads to diagonal stiff matrix and penta-diagonal mass matrix, and the resulting linear system can be solved in linear time. The above basis is equivalent to

$$\phi_j(\xi) = \frac{1-\xi}{2}\frac{1+\xi}{2} J_{j-2}^{1,1}(\xi) \tag{14}$$

with $J_j^{1,1}$ are the Jacobi polynomials that are orthogonal respect to weight $(1-\xi^2)$.

One may use different bases in different situations. When a lot of transform between spectral coefficients and physcial values are involved, Chebyshev method is a better choice, since fast transform is available. In other cases, Legendre bases might be the better choices. They are consistent with the weak formulation and resulting in very sparse system even in higher dimension problems.
If the degree of polynomial p is not very big in each element. One can also use *nodal basis*, i.e. the Lagrange basis:

$$\phi_j(\xi) = h_j(\xi), \quad j = 0, 1, ..., p. \tag{15}$$

where $\{h_j\}_{j=0}^p$ are Lagrange basis using Gauss-Legendre-Lobatto or Gauss-Chebyshev-Lobatto points, define in such order

$$\{\xi_0 = x_{i-1}, \xi_1 = x_i, \quad x_{i-1} < \xi_i < x_i, i = 2, ..., p\}. \tag{16}$$

Note that in this case, the boundary basis $h_0(\xi)$, $h_1(\xi)$ are not linear function but polynomials of degree p. Another difference to the *modal basis* is that the Lagrange basis leads to dense local mass and stiff matrix with a bad condition number even for equation with constant coefficients. This will increase the difficulty in solving the resulting linear algebraic system.

1.4 Solving the linear algebraic system

After the linear system (11) is built. There are two approach to solve it.

1. When p is small, but n is large, one can use well-known linear algebraic solver (e.g. PCG, AMG) to handle it.

2. When p is large, n is small, we use Schur complement.

Note that in Algorithm ASM, we have put all the boundary DoF in the very first. And we have $M^{i,i'}_{j,j'}=0$ and $S^{i,i'}_{j,j'}=0$, when $i\neq i'$ and $j,j'\neq 0,1$. This means the global matrix M and S has very special structures, which given as below

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where A_{11} is the vertex-vertex interaction part; A_{22} is the inner-inner interaction part, it is a block-diagonal matrix since inner DoF in different elements are not related. This means A_{22} is very easy to invert. To solve the linear system (11), is equivlent to solve a system

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{17}$$

or

$$A_{11}x_1+A_{12}x_2=b_1 \tag{18}$$

$$A_{21}x_1+A_{22}x_2=b_2 \tag{19}$$

From the second equation, we get $x_2=A_{22}^{-1}(b_2-A_{21}x_1)$, then plugging this into the first equation, we have

$$A_{11}x_1+A_{12}A_{22}^{-1}b_2-A_{12}A_{22}^{-1}A_{21}x_1=b_1$$

or

$$(A_{11}-A_{12}A_{22}^{-1}A_{21})x_1=b_1-A_{12}A_{22}^{-1}b_2. \tag{20}$$

This is a much smaller system. which can be solved more efficiently (for example using LU or PCG method). After solving for x_1, it is very easy to solve for x_2 using equation (19).

1.5 Error Estimates

1. By a standard approach (Céa Lemma), we can prove that, the solution error of the hp-Galerkin method, is bounded by projection error providing that the right side integration is evulated sufficiently accurate. Following is the sketch of the proof.
   Denote $a(u,v)=\alpha(u,v)+(u',v')$, then by taking difference of (2) and (4), we get

   $$a(u-u_N,v)=0,\quad \forall v\in X_0^{n,p}$$

   Let π_N is the projection from $H_0^1(I)\to X_0^{n,p}$, defined by

   $$a(u-\pi_N u,v)=0,\quad \forall v\in X_0^{n,p}.$$

   Since $u-u_N=(u-\pi_N u)+(\pi_N u-u_N)$, we have

   $$a(\pi_N u-u_N,v)=a(u-u_N,v)-a(u-\pi_N u,v)=-a(u-\pi_N u,v),\quad \forall v\in X_0^{n,p}$$

   Then by taking $v=\pi_N u-u_N\in X_0^{n,p}$, and using the continuous and coercive property of $a(\cdot,\cdot)$, we get

   $$\beta\|\pi_N u-u_N\|^2\leqslant a(\pi_N u-u_N,\pi_N u-u_N)<C\|u-\pi_N u\|\cdot\|\pi_N u-u_N\|,$$

   or

   $$\|\pi_N u-u_N\|\leqslant\frac{C}{\beta}\|u-\pi_N u\|.$$

Then by triangular inequality, we have

$$\|u-u_N\| \leqslant \|u-\pi_N u\| + \|\pi_N u - u_N\| \leqslant \left(1+\frac{C}{\beta}\right)\|u-\pi_N u\|.$$

2. Estimating the projection error. Since π_N is the projection defined by using the bilinear form $a(\cdot,\cdot)$ and the bilinear form induced energy norm is equivalent to the H^1 norm, so there exist a constant C', such that

$$\|u-u_N\| \leqslant C'\|u-\phi\|, \quad \forall\, \phi \in X_0^{n,p}.$$

In particular, for given $u \in H_0^1$, let take

$$\psi(x) = \sum_{i=1}^{n-1} u(x_i)\big(\psi_1^i(x) + \psi_0^{i+1}(x)\big), \quad \text{and} \quad \phi = \psi + \sum_{i=1}^{n} \pi_p^{0,i}(u(x)-\psi(x))|_{I_i},$$

where $\pi_p^{0,i}$ is the project from $H_0^1(I_i)$ to $P_N^0(I_i)$. By using ϕ, we have

$$\begin{aligned}
\|u-\phi\|^2 &= \|u'-\phi'\|^2_{L^2(\Omega)} + \|u-\phi\|^2_{L^2(I)} \\
&\leqslant C\|u'-\phi'\|^2_{L^2(\Omega)}, \qquad (\text{Poincare inequality}) \\
&= C\sum_{i=1}^{n} \|u'-\phi'\|^2_{L^2(I_i)} \\
&= C\sum_{i=1}^{n} \big\|u'-\psi'+[\pi_p^{0,i}(u-\psi)]'\big\|'_{L^2(I_i)} \qquad (21)
\end{aligned}$$

By using the Theorem 3.38 in [Shen J. et al 2011], it is easy to prove that

Lemma 1. *If $v \in H_0^1(I_i)$, and $v \in H^m(I_i), m \geqslant 1$, then we have*

$$\big\|\big(v-\pi_p^{0,i}v\big)'\big\|_{L^2(I_i)} \lesssim \left(\frac{|I_i|}{2}\right)^{m-1}\left(\sqrt{\frac{2}{e}}p\right)^{1-m}\|\partial_x^m\|_{L^2(I_i)}, \quad \forall\, p \geqslant m-1. \qquad \sharp$$

Taking $u-\psi = v$ in equation (21), we get

$$\|u-\phi\|^2 \leqslant C\sum_{i=1}^{n}\left(\frac{|I_i|}{2}\right)^{2(m-1)}\left(\sqrt{\frac{2}{e}}p\right)^{2(1-m)}\|\partial_x^m\|^2_{L^2(I_i)},$$

we further suppose that $|I_i| \equiv h = \frac{L}{n}$, then we get

$$\|u-\pi_N u\| \lesssim \|u-\phi\| \lesssim \left(\frac{2}{L}\sqrt{\frac{2}{e}}\,n\,p\right)^{1-m}\|\partial_x^m\|^c_{L^2(I)},$$

i.e.

$$\|u-u_N\| \lesssim \left(\frac{2}{L}\sqrt{\frac{2}{e}}\,n\,p\right)^{1-m}\|\partial_x^m\|_{L^2(I)}, \qquad \forall\, p+1 \geqslant m.$$

3. For $p+1<m$ (to be finished).

2 Spectral Element Method in Higher-Dimensions

The spectral element method in 2-dimensional domains and 3-dimensional domains are similar. For simplicity, we only consider the 2-dimensional case here. The model equation we will use is

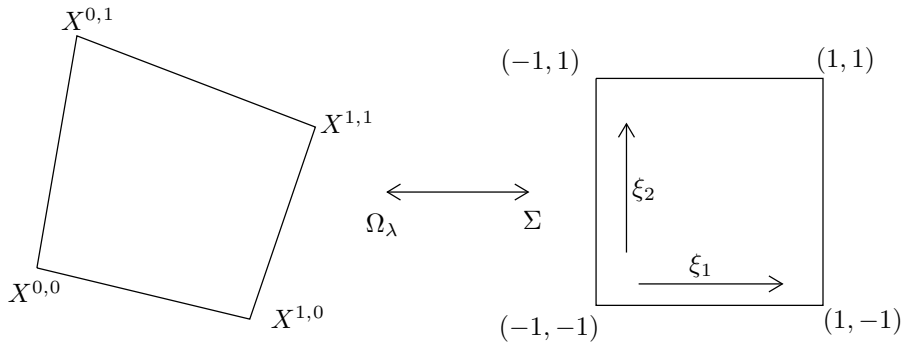
$$\begin{cases} \alpha\, u - \Delta u = f, \ \ (x_1,x_2) \in \Omega \\ u|_{\partial\Omega}=0. \end{cases} \qquad (22)$$

Here, we do not need Ω to be tensor product domain, since we can partition it into several small elements. These small elements can be quadrilateral elements, triangular elements or curved quadrangles or triangles. Here, we consider both quadrilateral and triangular elements. The weak form of equation (22) is

$$\begin{cases} \text{Find } u \in H_0^1(\Omega) \text{ such that} \\ \alpha(u,v) + (\nabla u, \nabla v) = (f,v), \quad \forall v \in H_0^1(\Omega). \end{cases} \tag{23}$$

2.1 Quadrilateral elements

2.1.1 Mapping



For a quadrilateral elements Ω_λ, we use local coordinates $(\xi_1, \xi_2) \in [-1,1]^2 =: \sum$ in reference (standard) domain. The global coordinates are denoted by $(x_1,\ x_2) \in \Omega_\lambda$. The mapping between local coorinates and global coordinates on element λ are given by

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = m_\lambda(\xi_1,\xi_2) = \sum_{i,j=0,1} \begin{pmatrix} x_1^{0,1} \\ x_2^{0,1} \end{pmatrix} \varphi_i(\xi_1)\varphi_j(\xi_2), \tag{24}$$

where $X^{i,j} = \left(x_1^{i,j},\ x_2^{i,j}\right)$ are the coordinates of four vertices; and the transform matrix is given by

$$\frac{\partial(x_1,x_2)}{\partial(\xi_1,\xi_2)} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \sum_{i,j} x_1^{i,j}\varphi_i'(\xi_1)\varphi_j(\xi_2) & \sum_{i,j} x_1^{i,j}\varphi_i(\xi_1)\varphi_j'(\xi_2) \\ \sum_{i,j} x_2^{i,j}\varphi_i'(\xi_1)\varphi_j(\xi_2) & \sum_{i,j} x_2^{i,j}\varphi_i(\xi_1)\varphi_j'(\xi_2) \end{pmatrix}. \tag{25}$$

The Jocobi determinant is given by

$$J = \begin{vmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} \\ \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2} \end{vmatrix} = \sum_{i,j=0,1} A^{i,j}\varphi_i(\xi_1)\varphi_j(\xi_2),$$

with

$$A^{i,j} = \frac{x_1^{1,j} - x_1^{0,j}}{2}\frac{x_2^{i,1} - x_2^{i,0}}{2} - \frac{x_1^{i,1} - x_1^{i,0}}{2}\frac{x_2^{1,j} - x_2^{0,j}}{2}$$

And also we have

$$\frac{\partial(\xi_1,\xi_2)}{\partial(x_1,x_2)} = \begin{pmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{pmatrix}^{-1} = \frac{1}{J}\begin{pmatrix} \frac{\partial x_2}{\partial \xi_2} & -\frac{\partial x_1}{\partial \xi_2} \\ -\frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_1} \end{pmatrix}.$$

Note that when the element is a rectangle or a parallelogram, the J determinant is a constant, in this case, stiff matrix can be made sparse by selecting special basis. Otherwise, J determinant is not a constant, the integral to calculate the components of stiff matrix are integration of rational functions, are hard to make sparse. And there are no simply formula to calculate these integration accurately. So we need resort to numerical integration to handle this.

2.1.2 Choices of bases

1. Modal basis. We choose $\{\phi_0=\varphi_0,\,\phi_1=\varphi_1\}\cup\{\phi_i,\,\phi_i(\pm1)=0\}_{i=2}^{p}$ as basis for both ξ_1 and ξ_2 direction. For Legendre case $\phi_i(\xi)=L_k(\xi)-L_{k-2}(\xi)$ for $k\geqslant 2$; while for Chebyshev case $\phi_i(\xi)=(1-\xi^2)T_{k-2}(\xi)$, $k\geqslant 2$. This will lead to sparse system for rectangle and parallelogram elements. We denote the basis in global variable as

$$\psi_{\boldsymbol{k}}^{\lambda}(x_1,x_2)=\begin{cases}\phi_{k_1}(\xi_1)\phi_{k_2}(\xi_2), & \text{if } (x_1,x_2)\in\Omega_\lambda\\ 0, & \text{otherwise}\end{cases}$$

Now let's calculate the local mass matrix and stiff matrix on element λ:

$$\begin{aligned}M_{\boldsymbol{k},\boldsymbol{k}'}^{\lambda} &= \int_{\Omega_\lambda}\psi_{\boldsymbol{k}}^{\lambda}(x_1,x_2)\psi_{\boldsymbol{k}'}^{\lambda}(x_1,x_2)\,\mathrm{d}x_1\mathrm{d}x_2\\ &= \int_{\Sigma}\phi_{k_1}(\xi_1)\phi_{k_2}(\xi_2)\phi_{k_1'}(\xi_1)\phi_{k_2'}(\xi_2)\,J\,\mathrm{d}\xi_1\mathrm{d}\xi_2\\ &= \sum_{i,j=0,1}A_{\lambda}^{i,j}\int_{-1}^{1}\phi_{k_1}(\xi_1)\phi_{k_1'}(\xi_1)\varphi_i(\xi_1)\mathrm{d}\xi_1\int_{-1}^{1}\phi_{k_2}(\xi_2)\phi_{k_2'}(\xi_2)\varphi_j(\xi_2)\,\mathrm{d}\xi_2\\ &= \sum_{i,j=0,1}A_{\lambda}^{i,j}\mathcal{M}_{k_1,k_1'}^{i}\mathcal{M}_{k_2,k_2'}^{j}\end{aligned}$$

with

$$\mathcal{M}_{l,l'}^{i}=\int_{-1}^{1}\phi_l(\xi)\phi_{l'}(\xi)\varphi_i(\xi)\,\mathrm{d}\xi,\,i=0,1.$$

On the other hand, the stiff matrix ($\partial_{x_1}^2$ part) are given by

$$\begin{aligned}S_{\boldsymbol{k},\boldsymbol{k}'}^{\lambda,1} &= \int_{\Omega_\lambda}\partial_{x_1}\psi_{\boldsymbol{k}}^{\lambda}(x_1,x_2)\partial_{x_1}\psi_{\boldsymbol{k}'}^{\lambda}(x_1,x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2\\ &= \int_{\Sigma}\left[\phi_{k_1}'(\xi_1)\phi_{k_2}(\xi_2)\frac{\partial\xi_1}{\partial x_1}+\phi_{k_1}(\xi_1)\phi_{k_2}'(\xi_2)\frac{\partial\xi_2}{\partial x_1}\right]\\ &\quad\times\left[\phi_{k_1'}'(\xi_1)\phi_{k_2'}(\xi_2)\frac{\partial\xi_1}{\partial x_1}+\phi_{k_1'}(\xi_1)\phi_{k_2'}'(\xi_2)\frac{\partial\xi_2}{\partial x_1}\right]J\,\mathrm{d}\xi_1\mathrm{d}\xi_2\\ &= \int_{\Sigma}\left[\phi_{k_1}'(\xi_1)\phi_{k_2}(\xi_2)\frac{\partial x_2}{\partial\xi_2}-\phi_{k_1}(\xi_1)\phi_{k_2}'(\xi_2)\frac{\partial x_2}{\partial\xi_1}\right]\\ &\quad\times\left[\phi_{k_1'}'(\xi_1)\phi_{k_2'}(\xi_2)\frac{\partial x_2}{\partial\xi_2}-\phi_{k_1'}(\xi_1)\phi_{k_2'}'(\xi_2)\frac{\partial x_2}{\partial\xi_1}\right]\frac{1}{J}\,\mathrm{d}\xi_1\mathrm{d}\xi_2\end{aligned}$$

   a. When J is a constant, both mass and stiff matrix are sparse. The nonzeros can be evaluated by using a Gauss quadrature. The matrix-vector product of mass matrix and stiff matrix can be evaluated efficiently using the tensor-product structure of the DoF on each element. e.g.

$$\begin{aligned}\sum_{\boldsymbol{k}}M_{\boldsymbol{k},\boldsymbol{k}'}^{\lambda}u_{\boldsymbol{k}}^{\lambda} &= \sum_{i,j=0,1}A_{\lambda}^{i,j}\left(\sum_{k_1,k_2=0}^{p}u_{k_1k_2}^{\lambda}\mathcal{M}_{k_1,k_1'}^{i}\mathcal{M}_{k_2,k_2'}^{j}\right)\\ &= \sum_{i,j=0,1}A_{\lambda}^{i,j}\left(\sum_{k_2=0}^{p}\left(\sum_{k_1=0}^{p}u_{k_1k_2}^{\lambda}\mathcal{M}_{k_1,k_1'}^{i}\right)\mathcal{M}_{k_2,k_2'}^{j}\right)\end{aligned}$$

b. When J is not constant, one can use a high-accurate Gauss quadrature to pre-calculate mass matrix and stiff matrix. However the stiff matrix $S^{\lambda,1}_{\boldsymbol{k},\boldsymbol{k}'}$ can be write as a sum of few tensor-product terms, since $1/J$ is not separable. In this case one can use pseudo-spectral approach. (Homeowrk: *design an efficient numerical scheme to numerically calculate the derivative of a function u, and further design a fast matrix-vector product algorithm*)

2. Nodal basis. One can use Lagrange basis $\{\phi_{j_1}(\xi_1)\phi_{j_2}(\xi_2)\}_{j_1,j_2=0}^{p}$ based no tensor product Gauss-Legendre-Lobatto or Gauss-Chebyshev-Lobatto points. The local mass matrix is a diagonal matrix

$$\begin{aligned}
M^{\lambda}_{\boldsymbol{k},\boldsymbol{k}'} &= \int_{\Omega_\lambda} \psi^{\lambda}_{\boldsymbol{k}}(\boldsymbol{x})\psi^{\lambda}_{\boldsymbol{k}'}(\boldsymbol{x})\,\mathrm{d}x_1\mathrm{d}x_2 \\
&= \int_{\Sigma} h_{k_1}(\xi_1)h_{k_2}(\xi_2)h_{k_1'}(\xi_1)h_{k_2'}(\xi_2)J\,\mathrm{d}\xi_1\mathrm{d}\xi_2 \\
&\cong \sum_{j_1,j_2=0}^{p} [h_{k_1}(\xi_1)h_{k_2}(\xi_2)h_{k_1'}(\xi_1)h_{k_2'}(\xi_2)J(\xi_1,\xi_2)]|_{\xi_1=\zeta_{j_1}}^{\xi_2=\zeta_{j_2}}\omega_{j_1}\omega_{j_2} \\
&= \sum_{j_1,j_2=0}^{p} \delta_{k_1 j_1}\delta_{k_2,j_2}\delta_{k_1' j_1}\delta_{k_2' j_2}J(\zeta_{j_1},\zeta_{j_2})\omega_{j_1}\omega_{j_2} \\
&= \delta_{\boldsymbol{k},\boldsymbol{k}'}J(\zeta_{k_1},\zeta_{k_2})\omega_{k_1}\omega_{k_2}
\end{aligned}$$

The stiff matrix is

$$\begin{aligned}
S^{\lambda,1}_{\boldsymbol{k},\boldsymbol{k}'} &= \int_{\Omega_\lambda} \partial_{x_1}\psi^{\lambda}_{\boldsymbol{k}}(\boldsymbol{x})\partial_{x_1}\psi^{\lambda}_{\boldsymbol{k}'}(\boldsymbol{x})\,\mathrm{d}x_1\mathrm{d}x_2 \\
&= \int_{\Sigma} [h'_{k_1}(\xi_1)h_{k_2}(\xi_2)\frac{\partial\xi_1}{\partial x_1} + h_{k_1}(\xi_1)h'_{k_2}(\xi_2)\frac{\partial\xi_2}{\partial x_1}] \\
&\quad \times [h'_{k_1'}(\xi_1)h_{k_2'}(\xi_2)\frac{\partial\xi_1}{\partial x_1} + h_{k_1'}(\xi_1)h'_{k_2'}(\xi_2)\frac{\partial\xi_2}{\partial x_1}]J\,\mathrm{d}\xi_1\mathrm{d}\xi_2 \\
&= \int_{\Sigma} \left[h'_{k_1}(\xi_1)h_{k_2}(\xi_2)\frac{\partial x_2}{\partial\xi_2} - h_{k_1}(\xi_1)h'_{k_2}(\xi_2)\frac{\partial x_2}{\partial\xi_1}\right] \\
&\quad \times \left[h'_{k_1'}(\xi_1)h_{k_2'}(\xi_2)\frac{\partial x_2}{\partial\xi_2} - h_{k_1'}(\xi_1)h'_{k_2'}(\xi_2)\frac{\partial x_2}{\partial\xi_1}\right]\frac{1}{J}\mathrm{d}\xi_1\mathrm{d}\xi_2
\end{aligned}$$

The each term of the integration can be approximated by Guass quadrature. For example,

$$\begin{aligned}
I^{\lambda,1}_{\boldsymbol{k},\boldsymbol{k}'} &= \int_{\Sigma} h'_{k_1}(\xi_1)h_{k_2}(\xi_2)\frac{\partial x_2}{\partial\xi_2}h'_{k_1'}(\xi_1)h_{k_2'}(\xi_2)\frac{\partial x_2}{\partial\xi_2}\frac{1}{J}\mathrm{d}\xi_1\,\mathrm{d}\xi_2 \\
&\approx \sum_{j_1=0}^{p}\sum_{j_2=0}^{p} h'_{k_1}(\zeta_{j_1})\delta_{k_2,j_2}h'_{k_1'}(\zeta_1)\delta_{k_2',j_2}\left(\frac{\partial x_2}{\partial\xi_2}(\zeta_{j_1})\right)^2\frac{1}{J(\zeta_{j_1},\zeta_{j_2})}\omega_{j_1}\omega_{j_2}
\end{aligned}$$

The corresponding matrix vector product is

$$\begin{aligned}
\sum_{\boldsymbol{k}} u^{\lambda}_{\boldsymbol{k}}I^{\lambda,1}_{\boldsymbol{k},\boldsymbol{k}'} &= \sum_{k_1,k_2=0}^{p} u^{\lambda}_{k_1k_2}\left(\sum_{j_1,j_2=0}^{p} h'_{k_1}(\zeta_{j_1})h'_{k_1'}(\zeta_{j_1})\left(\frac{\partial x_2}{\partial\xi_2}(\zeta_{j_1})\right)^2\delta_{k_2j_2}\delta_{k_2'j_2}\frac{\omega_{j_1}\omega_{j_2}}{J(\zeta_{j_1},\zeta_{j_2})}\right) \\
&= \sum_{j_1,j_2=0}^{p}\left(\left(\sum_{k_1,k_2=0}^{p} u^{\lambda}_{k_1k_2}h'_{k_1}(\zeta_{j_1})\delta_{k_2j_2}\right)\left(\frac{\partial x_2}{\partial\xi_2}(\zeta_{j_1})\right)^2\frac{\omega_{j_1}\omega_{j_2}}{J(\zeta_{j_1},\zeta_{j_2})}\right)h'_{k_1'}(\zeta_{j_1})\delta_{k_2'j_2}
\end{aligned}$$

This can be done in $O(p^3)$ operations by summing k_1,k_2 and j_1,j_2 successively.

2.1.3 Assembling and solving the linear system

The assembling procedure is similar to the 1-dimensional case. We use mapping function $\mathrm{map}(\lambda, k_1, k_2) = k$ to denote that the k_1-k_2-th local DoF in element λ has a global DoF index k. When making the DoF mapping, we put all the vertex DoF before edge DoF, and all edge DoF before inner DoF. Then the resulting linear system has a shape

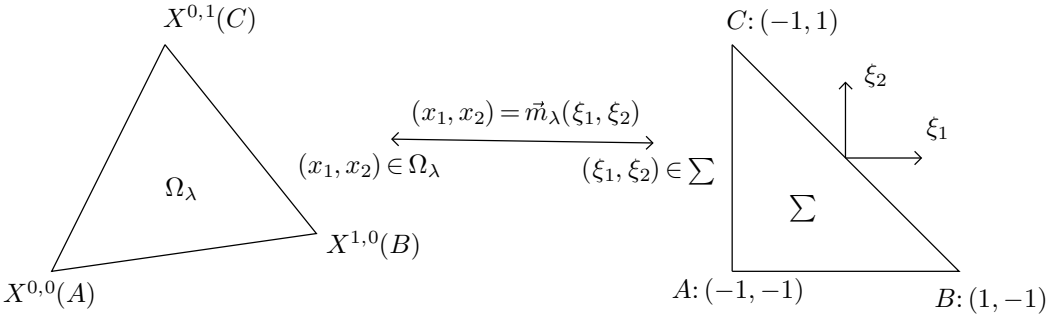
$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix},$$

where A_{11} is the vertex-vertex interaction part (has a shape of linear finite element matrix); A_{22} is the edge-edge interaction part; A_{33} is the inner-inter interaction part (block diagonal matrix). Let n be the number of elements, p be the the degree of polynomials in each element, then the size of A_{11}, A_{22}, A_{33} are $O(n), O(2n\,(p-1))$ and $O(n(p-1)^2)$.

1. When n is big, but p is small, we prefere using PCG or AMG method to solve the linear system.
2. When n is small, p is large, We can use Schur complement to solve the linear system.

2.2 Triangular Elements

2.2.1 Mapping from physical element to standard element



Suppose that the physical domain Ω, is partitioned to n triangular domain, i.e. $\Omega = \cup_{\lambda=0}^{n}\Omega_\lambda$. We first map each element Ω_λ, $\lambda = 1, \ldots, n$ to a standard reference tiangle $\sum = \{(\xi_1, \xi_2): -1 \leqslant \xi_1, \xi_2 \leqslant 1,\ \xi_1 + \xi_2 \leqslant 0\}$. Suppose the coordinates of the three vertex of Ω_λ are given by $X^{0,0}$, $X^{1,0}$, and $X^{0,1}$. Then the linear mapping function is

$$\binom{x_1}{x_2} = \varphi_1(\xi_1)X^{1,0} + \varphi_1(\xi_2)X^{0,1} + (1 - \varphi_1(\xi_1) - \varphi_1(\xi_2))X^{0,0}, \quad \varphi_1(\xi) = \frac{1+\xi}{2} \tag{26}$$

The Jacobi matrix is

$$J = \frac{\partial(x_1, x_2)}{\partial(\xi_1, \xi_2)} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \frac{x_1^{1,0} - x_1^{0,0}}{2} & \frac{x_1^{0,1} - x_1^{0,0}}{2} \\ \frac{x_2^{1,0} - x_2^{0,0}}{2} & \frac{x_2^{0,1} - x_2^{0,0}}{2} \end{pmatrix}$$

So the Jacobi determinant is a constant

$$|J| = \begin{vmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{vmatrix} = \frac{x_1^{1,0} - x_1^{0,0}}{2}\frac{x_2^{0,1} - x_2^{0,0}}{2} - \frac{x_2^{1,0} - x_2^{0,0}}{2}\frac{x_1^{0,1} - x_1^{0,0}}{2}.$$

The inverse of Jacobi matrix can be calculated as

$$\frac{\partial(\xi_1, \xi_2)}{\partial(x_1, x_2)} = \begin{pmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{pmatrix} = \frac{1}{|J|}\begin{pmatrix} \frac{x_2^{0,1} - x_2^{0,0}}{2} & -\frac{x_1^{0,1} - x_1^{0,0}}{2} \\ -\frac{x_2^{1,0} - x_2^{0,0}}{2} & \frac{x_1^{1,0} - x_1^{0,0}}{2} \end{pmatrix}.$$

Note that both the forward and backward Jacobi matrices are constant matrices, which means the local stiff and mass matrices on physical element and reference element has only a difference of constant matrices. i.e.

$$\int_{\Omega_\lambda} u(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{\sum} u(\xi_1,\xi_2)v(\xi_1,\xi_2)\,|J|\,\mathrm{d}\xi_1\mathrm{d}\xi_2$$

$$\begin{aligned}
&\int_{\Omega_\lambda} \partial_{x_1}u\partial_{x_1}v + \partial_{x_2}u\partial_{x_2}v\,\mathrm{d}x_1\mathrm{d}x_2\\
=\;& \int_{\sum}\left[u_1\frac{\partial\xi_1}{\partial x_1}+u_2\frac{\partial\xi_2}{\partial x_1}\right]\left[v_1\frac{\partial\xi_1}{\partial x_1}+v_2\frac{\partial\xi_2}{\partial x_1}\right]+\left[u_1\frac{\partial\xi_1}{\partial x_2}+u_2\frac{\partial\xi_2}{\partial x_2}\right]\left[v_1\frac{\partial\xi_1}{\partial x_2}+v_2\frac{\partial\xi_2}{\partial x_2}\right]|J|\,\mathrm{d}\xi_1\mathrm{d}\xi_2\\
=\;& \int_{\sum}\left[u_1\frac{\partial x_2}{\partial\xi_2}-u_2\frac{\partial x_2}{\partial\xi_1}\right]\left[v_1\frac{\partial x_2}{\partial\xi_2}-v_2\frac{\partial x_2}{\partial\xi_1}\right]+\left[u_2\frac{\partial x_1}{\partial\xi_1}-u_1\frac{\partial x_1}{\partial\xi_2}\right]\left[v_2\frac{\partial x_1}{\partial\xi_1}-v_1\frac{\partial x_1}{\partial\xi_2}\right]\frac{1}{|J|}\mathrm{d}\xi_1\mathrm{d}\xi_2\\
=\;& \frac{1}{|J|}\left\{A_{1,1}\int_{\sum}u_1v_1\,\mathrm{d}\xi_1\mathrm{d}\xi_2+A_{1,2}\int_{\sum}u_1v_2\,\mathrm{d}\xi_1\mathrm{d}\xi_2+A_{2,1}\int_{\sum}u_2v_1\,\mathrm{d}\xi_1\mathrm{d}\xi_2+A_{2,2}\int_{\sum}u_2v_2\,\mathrm{d}\xi_1\mathrm{d}\xi_2\right\}
\end{aligned}$$

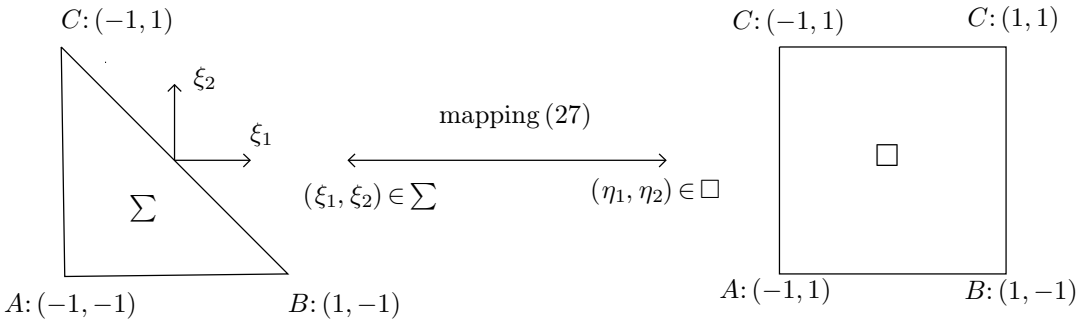
where $u_1=\partial_{\xi_1}u$, $u_2=\partial_{\xi_2}u$, and

$$A_{1,1}=\left(\frac{\partial x_2}{\partial\xi_2}\right)^2+\left(\frac{\partial x_1}{\partial\xi_2}\right)^2 \quad A_{2,2}=\left(\frac{\partial x_2}{\partial\xi_1}\right)^2+\left(\frac{\partial x_1}{\partial\xi_1}\right)^2 \quad A_{1,2}=A_{2,1}=-\frac{\partial x_2}{\partial\xi_1}\frac{\partial x_2}{\partial\xi_2}-\frac{\partial x_1}{\partial\xi_1}\frac{\partial x_1}{\partial\xi_2}\,.$$

2.2.2 Tensor product approximation using collapsed coordinates

On the standard reference element $\sum$, one can use tensor-product approximation or non-tensor-product approximation. Even though for the non-tensor product approximation usually use less DoF and have a more uniformly distributed grid, but tensor-product approximation has a faster matrix-vector algorithm. So in this lecture we only consider the tensor product based approximation. To use tensor-product approximation, we need map the standard triangular element to a square by using collapsed coordinates.

$$\begin{cases}\xi_1=\frac{(1+\eta_1)}{2}(1-\eta_2)-1,\\ \xi_2=\eta_2.\end{cases}\qquad (\eta_1,\eta_2)\in[-1,1]^2 \tag{27}$$



The Jacobi matrix of this transform is

$$\tilde{J}=\frac{\partial(\xi_1,\xi_2)}{\partial(\eta_1,\eta_2)}=\begin{pmatrix}\frac{\partial\xi_1}{\partial\eta_1} & \frac{\partial\xi_1}{\partial\eta_2}\\ \frac{\partial\xi_2}{\partial\eta_1} & \frac{\partial\xi_2}{\partial\eta_2}\end{pmatrix}=\begin{pmatrix}\frac{1-\eta_2}{2} & -\frac{1+\eta_1}{2}\\ 0 & 1\end{pmatrix}.$$

The Jacobi determinant is

$$\left|\tilde{J}\right|=\frac{1-\eta_2}{2}.$$

The Jacobi matrix of inverse transform is given by

$$\tilde{J}^{-1}=\begin{pmatrix}\frac{\partial\eta_1}{\partial\xi_1} & \frac{\partial\eta_1}{\partial\xi_2}\\ \frac{\partial\eta_2}{\partial\xi_1} & \frac{\partial\eta_2}{\partial\xi_2}\end{pmatrix}=\frac{1}{|\tilde{J}|}\begin{pmatrix}1 & \frac{1+\eta_1}{2}\\ 0 & \frac{1-\eta_2}{2}\end{pmatrix}=\begin{pmatrix}\frac{2}{1-\eta_2} & \frac{1+\eta_1}{1-\eta_2}\\ 0 & 1\end{pmatrix}.$$

Note that in the calculating of mass matrix, by using the collapsed coordinates transform, we will get an extra term $\left|\tilde{J}\right|$, which is $(1-\eta_2)/2$, is very easy to make sparse matrix. For the stiff matrix, we need to evaluate

$$\begin{aligned}
\int_{\sum} u_1 v_1 \,\mathrm{d}\xi_1 \mathrm{d}\xi_2 &= \int_{\square} (\partial_{\eta_1} u)(\partial_{\eta_1} v) \frac{2}{1-\eta_2} \mathrm{d}\eta_1 \mathrm{d}\eta_2 \\
\int_{\sum} u_1 v_2 \,\mathrm{d}\xi_1 \mathrm{d}\xi_2 &= \int_{\square} (\partial_{\eta_1} u)\left(\partial_{\eta_1} v \frac{1+\eta_1}{2} + \partial_{\eta_2} v \frac{1-\eta_2}{2} \right) \frac{2}{1-\eta_2} \mathrm{d}\eta_1 \mathrm{d}\eta_2 \\
\int_{\sum} u_2 v_2 \,\mathrm{d}\xi_1 \mathrm{d}\xi_2 &= \int_{\square} \left(\partial_{\eta_1} u \frac{1+\eta_1}{2} + \partial_{\eta_2} u \frac{1-\eta_2}{2} \right)\left(\partial_{\eta_1} v \frac{1+\eta_1}{2} + \partial_{\eta_2} v \frac{1-\eta_2}{2} \right) \frac{2}{1-\eta_2} \mathrm{d}\eta_1 \mathrm{d}\eta_2
\end{aligned}$$

When expanding $u(\eta_1, \eta_2)$ in tensor-product bases $\{\phi_j(\eta_1)\phi_k(\eta_2)\}$, we have those terms

$$\begin{aligned}
\int_{\square} (\phi_j'(\eta_1)\phi_k(\eta_2))(\phi_{j'}'(\eta_1)\phi_{k'}(\eta_2)) \frac{2}{1-\eta_2} \mathrm{d}\eta_1 \mathrm{d}\eta_2 &= s_{jj'} t_{kk'} \\
\int_{\square} (\phi_j'(\eta_1)\phi_k(\eta_2)) \left(\phi_{j'}'(\eta_1)\phi_{k'}(\eta_2) \frac{1+\eta_1}{1-\eta_2} + \phi_{j'}(\eta_1)\phi_{k'}'(\eta_2) \right) \mathrm{d}\eta_1 \mathrm{d}\eta_2 &= r_{jj'} t_{kk'} + q_{jj'} q_{k'k} \\
\int_{\sum} \partial_{\xi_2}(\phi_j(\eta_1)\phi_k(\eta_2)) \partial_{\xi_2}(\phi_{j'}(\eta_1)\phi_{k'}(\eta_2)) \,\mathrm{d}\xi_1 \mathrm{d}\xi_2 = a_{jj'} t_{kk'} + m_{jj'} b_{kk'} &+ c_{jj'} q_{k'k} + c_{j'j} q_{kk'}
\end{aligned}$$

where

$$\begin{aligned}
s_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}'(\eta) \,\mathrm{d}\eta, & t_{kk'} &= \int_{-1}^{1} \frac{2\phi_k(\eta)\phi_{k'}(\eta)}{1-\eta} \,\mathrm{d}\eta, \\
q_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}(\eta) \,\mathrm{d}\eta, & r_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}'(\eta) \frac{1+\eta}{2} \,\mathrm{d}\eta, \\
a_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}'(\eta) \left(\frac{1+\eta}{2} \right)^2 \mathrm{d}\eta & b_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}'(\eta) \frac{1-\eta}{2} \mathrm{d}\eta = s_{jj'} - r_{jj'}, \\
c_{jj'} &= \int_{-1}^{1} \phi_j'(\eta)\phi_{j'}(\eta) \frac{1+\eta}{2} \,\mathrm{d}\eta, & m_{jj'} &= \int_{-1}^{1} \phi_j(\eta)\phi_{j'}(\eta) \,\mathrm{d}\eta.
\end{aligned}$$

Note that only $t_{kk'}$ has a singular integration weight. To be integrable, $\phi_k(\eta)$ and $\phi_{k'}(\eta)$ should have a factor of $1-\eta$. The three linear bases associated with three vertices are given by

$$\psi_1 = \varphi_1(\xi_1) = \frac{1+\eta_1}{2}\frac{1-\eta_2}{2}, \quad \psi_2 = \varphi_1(\xi_2) = \frac{1+\eta_2}{2}, \quad \psi_0 = 1 - \varphi_1(\xi_1) - \varphi_2(\xi_2) = \frac{1-\eta_1}{2}\frac{1-\eta_2}{2}.$$

Like the polar coordinate transform. The expansion of function $u(\eta_1, \quad \eta_2)$ in terms of orthogonal polynomial basis in η_1 direction

$$u(\eta_1, \eta_2) = \sum_{k=0}^{\infty} c_k(\eta_2) L_k(\eta_1)$$

should satifies the ‘‘pole condition’’

$$c_k(\eta_2) \sim O\left(\left(\frac{1-\eta_2}{2} \right)^k \right), \quad \text{as} \quad \eta_2 \to 1. \tag{28}$$

This is the natural pole condition in the collapsed coorinate system. This is a corresponding essential pole condition similar to the polar cooridnate system

$$c_k(\eta_k) \sim O\left(\left(\frac{1-\eta_2}{2} \right) \right), \quad \text{as} \ \eta_2 \to 1, \quad \text{for} \quad k \geqslant 1. \tag{29}$$

Since only the first expansion coefficent $c_0(\eta_2)$ doesn't has a factor of $(1 - \eta_2)/2$, we need to verify that tha stiff matrix related to $c_0(\eta_2)L_0(\eta_1) = c_0(\eta_2)$ is integrable. Since $t_{jj'}$ always come with factor $s_{jj'}$ and $r_{jj'}$ which involve the integration of the derivatives of the bases and equal to zero for $L_0(\eta_1) \equiv 1$, so the integration is well-defined and equals to 0.

2.2.3 Choice of Basis

Here we only consider Modal bases. There are two choices.

1. Legendre basis with essential ‘‘pole condition’’. ψ_0, ψ_1, ψ_2 are boundary bases. Suppose the inner basis are tensor product type $\{\phi_i(\eta_1)\phi_j(\eta_2)\}_{i,j=2}^{p}$, where

$$\phi_j(\eta) = \frac{1-\eta^2}{4} J_{j-2}^{1,1}(\eta) = C_j\,(L_j(\eta) - L_{j-2}(\eta)).$$

A bases set on element Σ includes boundary bases, inner bases and also edge bases, which are given by

$$\begin{aligned}
\text{edge}\,\eta_2 = -1 \quad &: \quad \frac{1-\eta_2}{2}\,\phi_j(\eta_1), \quad j = 2, ..., p;\\
\text{edge}\,\eta_1 = -1 \quad &: \quad \frac{1-\eta_1}{2}\phi_j(\eta_2), \quad j = 2, ..., p;\\
\text{edge}\,\eta_1 = 1 \quad &: \quad \frac{1+\eta_1}{2}\phi_j(\eta_2), \quad j = 2, ..., p.
\end{aligned}$$

Note that thre are no bases on the collapsed edge: $\eta_2 = 1$ except the vertex basis $\psi_2 = \frac{1+\eta_2}{2}$. The overall approximation space is

$$\begin{aligned}
X_p \;&:=\; \{\, u(\eta_1,\eta_2) \in P_p \times P_p : u(\eta_1, 1) = u(-1, 1)\,\}\\
&=\; \text{span}\{\,\{\,\psi_0, \psi_1, \psi_2\} \cup \{\varphi_0(\eta_2)\phi_j(\eta_1), \varphi_0(\eta_1)\varphi_j(\eta_2), \varphi_1(\eta_1)\varphi_j(\eta_2)\}_{j=0}^{p}\\
&\qquad \cup \{\phi_i(\eta_1)\phi_j(\eta_2)\}_{i,j=2}^{p}\}
\end{aligned}$$

It is not too hard to prove that by using this kind of basis, we get sparse mass and stiff matrix. We can also use the tensor-product structure to do the fast matrix-vector multiplication.
One can also use the Lagrange bases based on tensor product G-L-L points.

2. Jaocib bases with natural ‘‘pole condition’’.

$$\phi_k(\eta_1) = L_k(\eta_1),\ k = 0, ..., p; \quad \phi_j^k(\eta_2) = \left(\frac{1-\eta_2}{2}\right)^k J_j^{2k+1,0}(\eta_2),$$

the basis set on element $\square$ is:

$$\{\,\psi_{kj}(\eta_1,\eta_2) = \phi_k(\eta_1)\phi_j^k(\eta_2), \quad k = 0, ..., p, j = 0, ..., p\,\}$$

or

$$\{\,\psi_{kj}(\eta_1,\eta_2) = \phi_k(\eta_1)\phi_j^k(\eta_2), \quad k = 0, ..., p, j = 0, ..., p-k\,\}.$$

Note that

$$\int_{\square} \psi_{kj}\psi_{k'j'}\frac{1-\eta_2}{2}\,\mathrm{d}\eta_1\mathrm{d}\eta_2 = \int_{-1}^{1} \phi_k(\eta_1)\phi_{k'}(\eta_1)\,\mathrm{d}\eta_1 \int_{-1}^{1} \phi_j^k(\eta_2)\phi_{j'}^{k'}(\eta_2)\frac{1-\eta_2}{2}\,\mathrm{d}\eta_2 = \frac{\delta_{kk'}\delta_{jj'}}{2^{2k+1}}$$

so this kind of basis sets is also called L^2-orthogonal bases. However, the stiff matrix is not sparse.

Remark 2. It is possible to build bases satisfies ‘‘pole condition’’but also produce sparse matrices. For example: $u = c_0 + \sum_{k=0}^{p}\sum_{j=0}^{p_k} u_{kj}\phi_k(\eta_1)\phi_j^k(\eta_2)$, where $\{\phi_k\}_{k=0}^{p} = \left\{\frac{1-\eta}{2}, \frac{1+\eta}{2}, \frac{1-\eta^2}{4}J_{k-2}^{1,1}, k = 2, ..., p\right\}$ and

$$\phi_j^k = \begin{cases}
\frac{1-\eta_2}{2}, & k = 0, 1; \quad j = 0,\\
\frac{1-\eta_2}{2}\frac{1+\eta_2}{2}L_{j-1}(\eta_2), & k = 0, 1; \quad j = 1, ..., p_k,\\
\left(\frac{1-\eta_2}{2}\right)^k, & k = 2, ..., p; \quad j = 0,\\
\left(\frac{1-\eta_2}{2}\right)^k\frac{1+\eta_2}{2}\,J_{j-1}^{2k-4,0}(\eta_2), & k = 2, ..., p; \quad j = 1, ..., p_k.
\end{cases} \qquad p_k = \begin{cases} p-1, & k = 0, 1,\\ p-k, & k = 2, ..., p. \end{cases}$$