1. Assignment 1

1.1. 1-D

We consider the "*unit square*" in 1-D, $\xi = [-1, 1]$. Given two functions,

$$g(\xi) = \sin(\pi\xi) \ ,$$

$$f(\xi) = \pi \cos(\pi\xi) \ .$$

Let $-1 = \xi_0 < \xi_1 < \cdots < \xi_N = 1$ be $N+1$ Gauss-Lobatto nodes. They give rise to $N+1$ Lagrange basis functions of degree N:

$$h_i(\xi), \quad i = 0, 1, \cdots, N.$$

Meanwhile, N edges, $[\xi_{i-1}, \xi_i]$, $i = 1, 2, \cdots, N$, result in N edge basis functions of degree $N-1$:

$$e_i(\xi), \quad i = 1, 2, \cdots, N.$$

What we have to do is to discretize $g(\xi)$ with $h_i(\xi)$ into

$$g^h(\xi) := \sum_{i=0}^{N} g_i h_i(\xi) \ ,$$

where $g_i = g(\xi_i)$, and to discretize $f(\xi)$ with $e_i(\xi)$ into

$$f^h(\xi) := \sum_{i=1}^{N} f_i e_i(\xi) \ ,$$

where $f_i = \int_{\xi_{i-1}}^{\xi_i} f(\xi) \mathrm{d}\xi$.

Lets call the process of getting the coefficients, g_i, f_i, from the analytical functions **reduction**, and call the process of obtaining $g^h(\xi)$ and $f^h(\xi)$ from the coefficients **reconstruction**. They are what you have to code in this assignment.

1.2. 2-D

Once you have successfully done the 1-D case, I believe you now are able to do the 2-D case. We consider the real "*unit square*", $\xi \times \eta = [-1, 1]^2$, and have a vector $\boldsymbol{u} = [u(\xi, \eta), v(\xi, \eta)]^T$ and a scalar $f(\xi, \eta)$ given as

$$\begin{cases} u(\xi, \eta) = \sin(\pi\xi)\cos(\pi\eta) \\ v(\xi, \eta) = \cos(\pi\xi)\sin(\pi\eta) \end{cases} ,$$

$$f(\xi, \eta) = 2\pi \cos(\pi\xi)\cos(\pi\eta) \ .$$

Now, we not only construct basis functions $h_i(\xi)$ and $e_i(\xi)$ along ξ-axis like what we have done in 1-D case, but also have to construct basis function, $h_j(\eta)$, $e_j(\eta)$, along η-axis.

With these basis functions, we can discretize $f(\xi, \eta)$ either into

$$f^h(\xi, \eta) := \sum_{i=0}^{N} \sum_{j=0}^{N} f_{i,j} h_i(\xi) h_j(\eta) \ ,$$

where $f_{i,j} = f(\xi_i, \eta_j)$, or into

$$f^h(\xi, \eta) := \sum_{i=1}^{N} \sum_{j=1}^{N} f_{i,j} e_i(\xi) e_j(\eta) \ ,$$

where $f_{i,j} = \int_{\xi_{i-1}}^{\xi_i} \int_{\eta_{j-1}}^{\eta_j} f(\xi, \eta) \mathrm{d}\xi \mathrm{d}\eta$.

As for the vector function $\boldsymbol{u} = [u(\xi, \eta), v(\xi, \eta)]^T$, we discretize its components with basis functions $h_i(\xi) e_j(\eta)$ and $e_i(\xi) h_{j(\eta)}$ into

$$\begin{cases} u^h(\xi, \eta) := \displaystyle\sum_{i=0}^{N} \sum_{j=1}^{N} u_{i,j} h_i(\xi) e_j(\eta) \\ v^h(\xi, \eta) := \displaystyle\sum_{i=1}^{N} \sum_{j=0}^{N} v_{i,j} e_i(\xi) h_j(\eta) \end{cases} ,$$

where $u_{i,j} = \int_{\eta_{j-1}}^{\eta_j} u(\xi_i, \eta) \mathrm{d}\eta$ and $v_{i,j} = \int_{\xi_{i-1}}^{\xi_i} v(\xi, \eta_j) \mathrm{d}\xi$.

These more or less concludes the processes of **reduction** and **reconstruction** in 2-D.

Please code your program doing these 2-D **reduction** and **reconstruction** and play with it using different N, or even you can have different N, i.e. N_x and N_y, along ξ-axis and η-axis.