Discrete Geometries of Mathematics and Physics

Mimetic spectral element method[1]

Assignment #5

Weak formulation and discretization: Multiple elements

Yi Zhang (张仪)
: www.mathischeap.com
@: zhangyi_aero@hotmail.com
git: https://github.com/mathischeap

In Assignment #4, we have studied and programmed the application of the mimetic spectral element method to the Poisson equation in the computational domain $\Omega = [-1,1]^2$. However, there we consider the whole computational domain Ω to be a single physical element. Thus, we only need to construct one mapping

$$\Phi : \Omega_{\mathrm{r}} \to \Omega$$

for the discretization. This is fine for this particular case as the comptuational domain Ω is a regular domain. However, in most cases, we may face a irregular domain that we cannot cover it with a single element, and we have to divide the domain into multiple elements. This is the well-known *mesh generation*. There are in fact other reasons which drive us to use more than one element. For now, we leave them for your own after-class thinking and reading. In this assignment, we study how to apply the mimetic spectral element method on a mesh of multiple elements to the Poisson problem as in Assignment #4. We use the domain Ω as a demonstration.

1 Mesh generation

We consider the same computational domain $\Omega = [-1,1]^2$. Let K be a positive integer. In this domain, we generate a mesh of K^2 uniform elements. In other words, along each axis the domain is divided into K elements. So each element is a small square (a special orthogonal rectangle) whose edge length is $h = \frac{1}{K}$. And we use Ω_{ij} to denote the element

$$\Omega_{ij} := [h(i-1), hi] \times [h(j-1), hj], \quad i,j \in \{1, 2, \cdots, K\}.$$

And these elements are globalled numbered as

$$\Omega_k = \Omega_{(j-1)\times K + i} = \Omega_{ij}, \quad i,j \in \{1, 2, \cdots, K\}.$$

[1] https://mathischeap.com/contents/teaching/advanced_numerical_methods/mimetic_spectral_element_method/main

So, k ranges from 1 to K^2. See Fig. 1 for an example of this mesh.

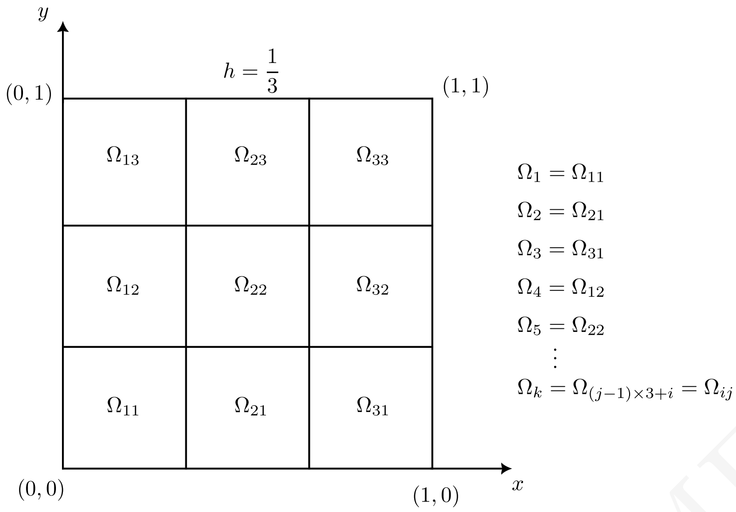

Fig. 1: An illustration of the mesh at $K = 3$.

Now, for an element Ω_k, we can construct a mapping which maps the reference element into it, i.e.,

$$\Phi_k : \Omega_{\mathrm{r}} \to \Omega_k$$

by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \Phi_k(\xi, \eta) = \begin{bmatrix} \Phi_k^x(\xi) \\ \Phi_k^y(\eta) \end{bmatrix} = \begin{bmatrix} h(\xi + 1)/2 + h(i-1) \\ h(\eta + 1)/2 + h(j-1) \end{bmatrix}.$$

According to what we have learned in Assignment #2, we can quickly obtain the metric-related values for the element,

$$\boldsymbol{\mathcal{J}}_k = \begin{bmatrix} \frac{h}{2} & 0 \\ 0 & \frac{h}{2} \end{bmatrix},$$

$$J_k = \det(\boldsymbol{\mathcal{J}}_n) = \frac{h^2}{4},$$

$$\boldsymbol{\mathcal{G}}_k = \begin{bmatrix} \frac{h^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix},$$

$$g_k = \det(\boldsymbol{\mathcal{G}}_k) = J_n^2 = \frac{h^4}{16},$$

$$\boldsymbol{\mathcal{J}}_k^{-1} = \begin{bmatrix} \frac{2}{h} & 0 \\ 0 & \frac{2}{h} \end{bmatrix},$$

$$J_k^{-1} = \frac{4}{h^2},$$

$$\boldsymbol{\mathcal{G}}_k^{-1} = \begin{bmatrix} \frac{4}{h^2} & 0 \\ 0 & \frac{4}{h^2} \end{bmatrix}.$$

Obviously, it is very convenient to do reduction and reconstruction and to compute mass matrices using this mapping the its metric-related values. And, because $h = \frac{1}{K}$ is same for all elements, the metric-related values are same for all elements.

2 Discretization of the Poisson problem in elements

In Assignment #4, we do the discretization on the whole domain Ω because we consider it as one element. Now, we do the discretization in an element Ω_k. Obviously, the resulting system is similar. In Ω_k, we can obtain the following linear system,

$$\begin{bmatrix} \mathbb{M}_D^h & \mathbb{E}_D^\mathsf{T}\mathbb{M}_S^h \\ \mathbb{E}_D & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{\boldsymbol{u}}_k \\ \vec{\varphi}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\vec{f}_k \end{bmatrix}. \tag{1}$$

We can see that the incidence matrix does not change. And the mass matrices becomes different because they depend on h. And we use $\vec{\boldsymbol{u}}_k, \vec{\varphi}_k, \vec{f}_k$ to represent the vector of expansion coefficients of variables in element Ω_k. Note that, in Assignment #4, we used the homogeneous boundary condition $\varphi = 0$ on $\partial\Omega$ to obtain a system similar to (1). However, if Ω_k is, for example, an internal element, it does not have a similar boundary condition. Thus, ideally, (1) should have terms corresponding to boundary integral. We have omitted them since the contribution of these boundary integrals from different elements will cancel each other eventually.

For $k \in \{1, 2, \cdots, K^2\}$, we can do the discretization and obtain a local system

$$\boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{b}_k. \tag{2}$$

As we have analyzed, the left-hand size matrix $\boldsymbol{A}$ will be same for all elements, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} \mathbb{M}_D^h & \mathbb{E}_D^\mathsf{T}\mathbb{M}_S^h \\ \mathbb{E}_D & \mathbf{0} \end{bmatrix}, \tag{3}$$

And $\boldsymbol{x}_k = \begin{bmatrix} \vec{\boldsymbol{u}}_k \\ \vec{\varphi}_k \end{bmatrix}$, $\boldsymbol{b}_k = \begin{bmatrix} \mathbf{0} \\ -\vec{f}_k \end{bmatrix}$ will be different from elementt to element.

This means we will have K^2 local linear systems for the K^2 elements. Clearly, we are not going to solve them individually; they belong to one complete problem. We somehow need to assemble these K^2 local linear systems into one global system and solve the global system. This leads to an essential topic of all finite element methods, *assembling*. We will address it in the next two sections.

3 Global labeling: Gathering matrix

The key issue is: How to assemble the K^2 local linear systems for the K^2 elements into a global system? To do this, we need to first do the global labeling of all expansion coefficients (or degrees of freedom, DoF's).

Remember that the expansions coefficients (and the corresponding basis functions) are labeled locally in an element such that we can make the incidence and mass matrices. See Fig.2 and Fig.3 for examples of local labeling.

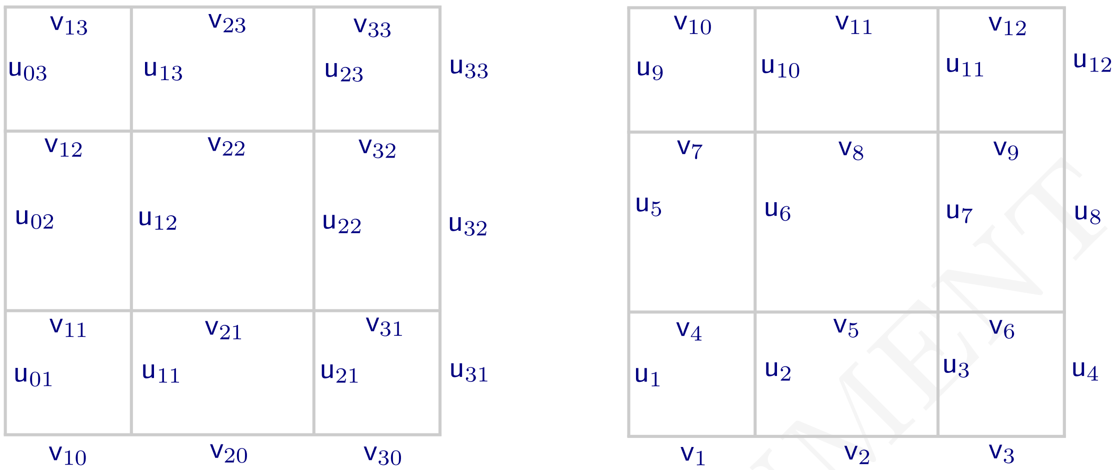

Fig. 2: For an element of $N = 3$. Left: Distribution of expansion coefficients u_{ij} and v_{ij} of space $\boldsymbol{D}(\Omega_k)$. Right: Local labeling of expansion coefficients u_{ij} and v_{ij} using $\mathsf{u}_m = \mathsf{u}_{(j-1)\times(N+1)+i+1} = \mathsf{u}_{ij}$ and $\mathsf{v}_m = \mathsf{v}_{j\times N+i} = \mathsf{v}_{ij}$.

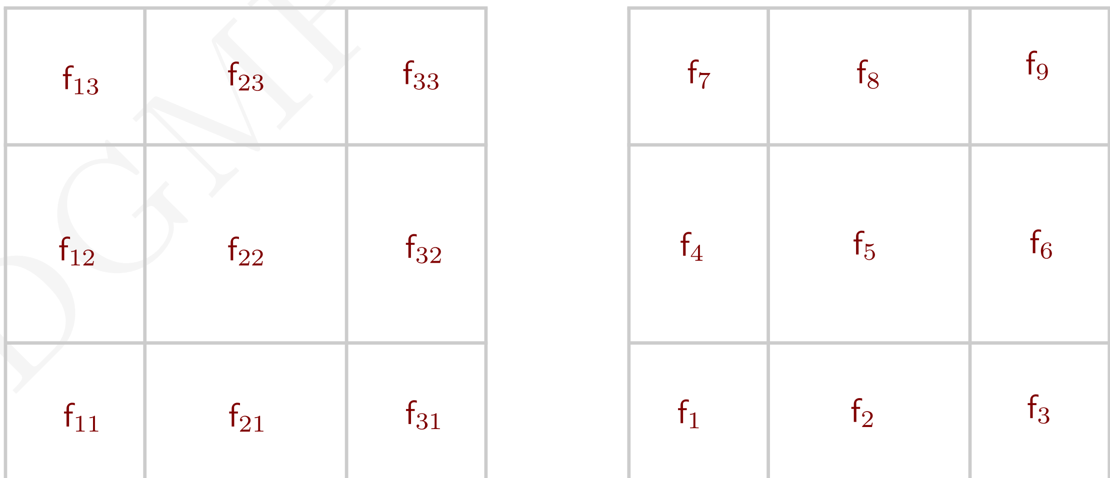

Fig. 3: For an element of $N = 3$. Left: Distribution of expansion coefficients f_{ij} of space $S(\Omega_k)$. Right: Local labeling of expansion coefficients f_{ij} using $\mathsf{f}_m = \mathsf{f}_{(j-1)\times N+i} = \mathsf{f}_{ij}$.

The local labeling happens locally in each element. So it is same in all elemenets. The global labeling is different. The global labeling labels all DoF's globally such that each different DoF has one unique label. Even if a DoF is shared by two elements, it must only have one label. For example, if two element are attached through their left-right edges. For example, see Fig. fig: local labeling D. The DoF $\mathsf{u}_{31} = \mathsf{u}_4$ of the left element is in fact coincident with the Dof $\mathsf{u}_{01} = \mathsf{u}_1$ of the right element. An example of a global labeling is shown in Fig. 4.

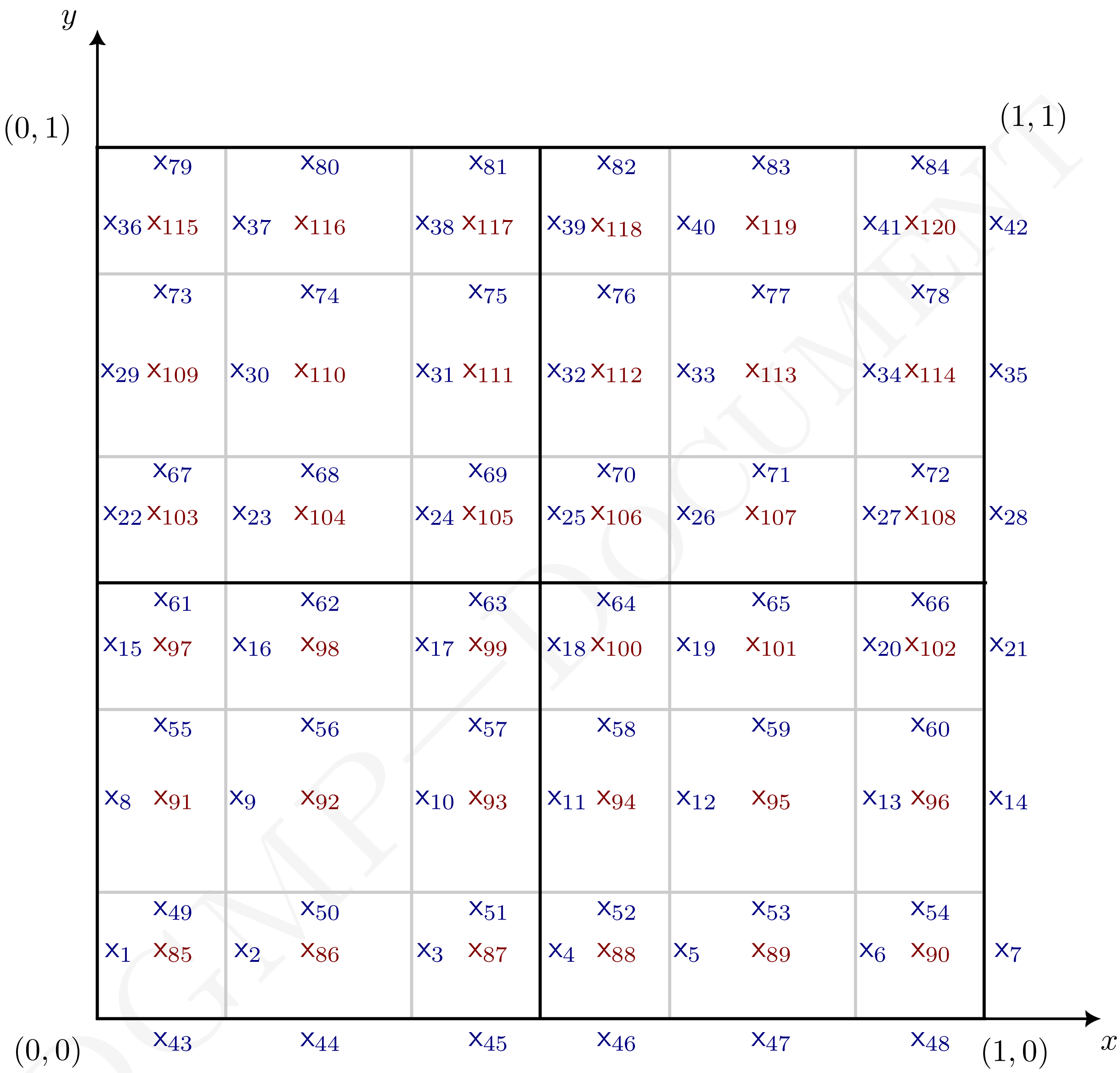

Fig. 4: A global labeling for $K = 2$ and $N = 3$.

In Fig. 4, a global labeling for $K = 2$ and $N = 3$ is presented. Since we use $K = 2$, so the domain is divided into 4 elements. The left-bottom element is Ω_1; the right-bottom one is Ω_2; the left-top one is Ω_3; the right-top one is Ω_4.

In this mesh of 4 elements, we can see that in total we have 120 DoF's. Among them, 84 (colored blue) are DoF's of $\boldsymbol{u}_h$ and 36 (colored red) of them are DoF's of φ_h. They are distributed in the 4 elements. And some DoF's of $\boldsymbol{u}_h$ are shared by two elemenets. Note that since the local system is going to be assembled into one global system, we have named the DoF's by x_i.

Under this global labeling, we can obtain the so-called gathering matrix $\mathbb{G}$ for the mesh. Because we have 4 elements and in each element we have $24 + 9 = 33$ DoF's ($33 \times 4 > 120$ because some DoF's are shared by elements), the gathering matrix $\mathbb{G}$ is a 4×33 matrix, see it on the next page.

$$\mathbb{G} = \begin{bmatrix} 1 & 2 & 3 & 4 & 8 & 9 & 10 & 11 & 15 & 16 & 17 & 18 & 43 & 44 & 45 & 49 & 50 & 51 & 55 & 56 & 57 & 61 & 62 & 63 & 85 & 86 & 87 & 91 & 92 & 93 & 97 & 98 & 99 \\ 4 & 5 & 6 & 7 & 11 & 12 & 13 & 14 & 18 & 19 & 20 & 21 & 46 & 47 & 48 & 52 & 53 & 54 & 58 & 59 & 60 & 64 & 65 & 66 & 88 & 89 & 90 & 94 & 95 & 96 & 100 & 101 & 102 \\ 22 & 23 & 24 & 25 & 29 & 30 & 31 & 32 & 36 & 37 & 38 & 39 & 61 & 62 & 63 & 67 & 68 & 69 & 73 & 74 & 75 & 79 & 80 & 81 & 203 & 104 & 105 & 109 & 110 & 111 & 115 & 116 & 117 \\ 25 & 26 & 27 & 28 & 32 & 33 & 34 & 35 & 39 & 40 & 41 & 42 & 64 & 65 & 66 & 70 & 71 & 72 & 76 & 77 & 78 & 82 & 83 & 84 & 106 & 107 & 108 & 112 & 113 & 114 & 118 & 119 & 120 \end{bmatrix}.$$

To understand this gathering matrix, we first go back the local linear system that is in the following format, see (2),

$$\boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{b}_k,$$

where $\boldsymbol{x} = \begin{bmatrix} \vec{\boldsymbol{u}}_k \\ \vec{\varphi}_k \end{bmatrix}$. And from the local labeling, we know in each element

$$\vec{\boldsymbol{u}}_k = \begin{bmatrix} \mathsf{u}_1 \\ \mathsf{u}_2 \\ \mathsf{u}_3 \\ \vdots \\ \mathsf{u}_{12} \\ \mathsf{v}_1 \\ \mathsf{v}_2 \\ \mathsf{v}_3 \\ \vdots \\ \mathsf{v}_{12} \end{bmatrix}, \quad \vec{\varphi}_k = \begin{bmatrix} \mathsf{f}_1 \\ \mathsf{f}_2 \\ \mathsf{f}_3 \\ \vdots \\ \mathsf{f}_{N^2} \end{bmatrix}.$$

Therefore, we know

$$\vec{\boldsymbol{x}}_k = \begin{bmatrix} \mathsf{u}_1 \\ \mathsf{u}_2 \\ \mathsf{u}_3 \\ \vdots \\ \mathsf{u}_{12} \\ \mathsf{v}_1 \\ \mathsf{v}_2 \\ \mathsf{v}_3 \\ \vdots \\ \mathsf{v}_{12} \\ \mathsf{f}_1 \\ \mathsf{f}_2 \\ \mathsf{f}_3 \\ \vdots \\ \mathsf{f}_9 \end{bmatrix}.$$

And if we take the first element (row index 1), i.e., the left-bottom element, as an example. The first (colume index 1) local DoF u_1 is globally labeled x_1, see Fig. 2 and Fig. 4. Therefore, we have

$$\mathbb{G}|_{1,1} = 1.$$

And The 5th (colume index 5) local DoF u_5 is globally labeled x_8. Therefore,

$$\mathbb{G}|_{1,5} = 8.$$

And the last (33rd) (colume index 33) local DoF f_9 is globally labeled x_{99}. Therefore,

$$\mathbb{G}|_{1,33} = 99.$$

Similarly, you can understand all entries in $\mathbb{G}$.

4 Assembling

We still take the global labeling in Fig. 4 as an example. From thise global labeling, we know the global system must be a system of shape 120×120. And we now define the global system to be

$$\mathbb{A}\mathbb{X} = \mathbb{B},$$

where $\mathbb{A}$ is a 120×120 square matrix, $\mathbb{X}$ is a 120×1 colume vector, and $\mathbb{B}$ is a 120×1 colume vector. We just need to assemble $\mathbb{A}$ and $\mathbb{B}$ and send them to a linear solver. It will solve for $\mathbb{X}$.

To assemble $\mathbb{A}$, we first initialize $\mathbb{A}$ as an empty (zero) matrix,

$$\mathbb{A} = \mathbf{0}|_{120\times 120}.$$

Then we do

```
for k in range(0, 4):
    for i in range(0, 120):
        row = G[k, i]
        for j in range(0, 120):
            col = G[k, j]
            global_A[row, col] += local_A[i, j]
```

where `global_A` represents $\mathbb{A}$ and `local_A` represents $\boldsymbol{A}$, see (3).

To assemble $\mathbb{B}$, we first initialize $\mathbb{B}$ as an empty (zero) vector,

$$\mathbb{B} = \mathbf{0}|_{120\times 1} .$$

Then we do

```python
for k in range(0, 4):
    for i in range(0, 120):
        row = G[k, i]
        global_B[row] += local_b[k][i]
```

where `global_B` represents $\mathbb{B}$ and `local_b[k]` is $\boldsymbol{b}_k$.

After assembling $\mathbb{A}$ and $\mathbb{B}$, we send them to a linear solver and solve for $\mathbb{X}$. Once we obtain $\mathbb{X}$, we need to distribute entries of $\mathbb{X}$ to $\boldsymbol{x}_k$ by

```python
for k in range(0, 4):
    for i in range(0, 120):
        row = G[k, i]
        x[k][i] = global_X[row]
```

where `global_X` represents $\mathbb{X}$ and `x[k]` is $\boldsymbol{x}_k$.

later, we can obtain $\boldsymbol{u}_k$ and φ_k from $\boldsymbol{x}_k$. With $\boldsymbol{u}_k$ and φ_k, we can finally reconstruct solutions $\boldsymbol{u}_h$ and φ_h in each elements.

Assignment 5.1.0: Program it!

Program it to solve the same Poisson problem as in Assigment 4.1.0, but this time with a mesh of K^2 uniform elements. Reconstruct the solutions $\boldsymbol{u}_h$ and φ_h to valid your program.

Try it using, for example, $N \in \{2, 3, 4\}$ and $K \in \{4, 8, 12, 16\}$.