THE CHINESE UNIVERSITY OF HONG KONG

PROGRESS SUMMARY

Zhe Zhu, Yin Ki Ng
August 5, 2016

1 ABSTRACT

This project is based on the result of predecessor's project of 1D-DG method and is going to explore how we may construct the parallel code for DG-FEM to solve PDE. The 1D,2D and 3D case of the Poisson 's equation will be discussed respectively. After that, we will see whether we can extend the same method to some time-dependent equation.

Some background information of FEM and DG will be given in Section 2. The idea of FEM and an example of it will be given in Section 2.1 and 2.2. The implement of FEM in 1D Poisson's equation will be showed in Section 2.3 and 2.4. The general dimensional implement of FEM will be given in Section 2.5

The Section 3 introduce the detail of how to construct the stiffness matrix. Section 3.1 introduce the main idea of the construction briefly. Section 3.2 and 3.3 introduce the idea of mapping. Section 3.4, 3.5, 3.6 introduce the calculation of each term.

The Section 4, 5, 6 gives examples of solving Poisson's equation in 1D, 2D and 3D.

The future work is discussed finally in Section 6

2 INTRODUCTION OF DG-FEM

Overview: In this section, the basic idea of Finite Element Method and Discountinous Galerkin Method will be introduced.

2.1 BASIC INFORMATION

Consider solving an PDE of the form

$$Lu(x) = f(x), \quad \forall x \in \Omega \subset R^m \tag{2.1}$$

where L is a linear differential operator of second order, and f is a given function. A typical example of the operator L that arises from many physical and engineering applications is given by

$$Lu(x) = -\triangle \cdot (q(x)\triangle u) + c(x)u, \quad \forall x \in \Omega \subset R^m \tag{2.2}$$

Both mathematically and physically the solution u should satisfy some boundary conditions in order for the differential equation (2.1) to be well defined. There are three typical boundary conditions:

1. Dirichlet boundary condition:

$$u = 0 \quad on \quad \Omega \tag{2.3}$$

2. Neumann boundary condition:

$$\frac{\partial u}{\partial \mathbf{n}} = 0 \quad on \quad \Omega \tag{2.4}$$

3. Robin boundary condition:

$$\frac{\partial u}{\partial \mathbf{n}} + \alpha(x)u = 0 \quad on \quad \Omega \tag{2.5}$$

where $\alpha(x)$, $q(x)$ and $c(x)$ are given function in Ω.

For our subsequent formulation, we introduce the following spaces

$$\begin{aligned} L^2(\Omega) &= \{w; \int_\Omega w^2(x)dx < \inf\} \\ H^1(\Omega) &= \{w \in L^2(\Omega); \int_\Omega w^2(x) + \sum_{i=1}^{m} w_{x_i}^2(x))dx < \inf\} \\ H_0^1(\Omega) &= \{w \in H^1(\Omega); w = 0 \quad on \quad \partial u\} \end{aligned} \tag{2.6}$$

And for all $u, v \in L^2(\Omega)$, we define the scalar product

$$(u, v)_\Omega = \int_\Omega u(x)v(x)dx \tag{2.7}$$

Associated with the operator L, we introduce a bilinear form

$$a(u, v) = (Lu, v), \quad \forall u, v \in H^1(\Omega). \tag{2.8}$$

Using the integration by parts formula:

$$\int_\Omega u_{x_i} v dx = -\int_\Omega u v_{x_i} dx + \int_{\partial\Omega} u v n_i dx \tag{2.9}$$

where $\mathbf{n} = (n_1, n_2, ..., n_m)^T$ is the outward normal unitary direction to $\partial\Omega$. We can usually assume that $a(u, v)$ is symmetric and positive definite.

We now try to find an approximate solution of u(x) to the differential equation (2.1), with some boundry conditions, such as Dirichlet boundary condition (2.3). To do so, we choose a set of basis functions

$$V_h = span\{\phi 1, \phi 2, ..., \phi_n\} \subset H_0^1(\Omega) \tag{2.10}$$

and then try to find an approximate solution u_n of u in the form

$$u_n(x) = \sum_{j=1}^{n} \alpha_j \phi_j(x) \tag{2.11}$$

where the coefficients $\alpha_1, \alpha_2, ..., \alpha_n$ are to be determined. This is what FEM try to solve.

2.2 Finite Element Method (FEM)

Consider the following 1D-problem, involving a differential equation with boundary conditions:

$$\begin{cases} -u'' = f \\ u(a) = u(b) = 0 \end{cases} \tag{2.12}$$

We wish to solve the unknown **trial** function u on the domain $I = [a, b]$. As mentioned before, we may do so by multiplying an arbitrary **test** function v defined such that it has the same boundary condition $v(a) = v(b) = 0$ as u. We integrate over the domain to get

$$-\int (u'' + f)v = -\int u''v - \int fv = 0 \tag{2.13}$$

Suppose we choose the test function v to be continuous and differentiable everywhere on the domain. Integration by part on the first term gives

$$\int u'v' - u'v|_a^b - \int fv = 0$$

The central term disappears so we have

$$\int u'v' - \int fv = 0 \tag{2.14}$$

Notice that instead of having a twice derivative term in equation (2.13) (**strong form**), now we get a first derivative term in equation (2.14) (**weak form**). Cut the domain I into intervals $I_0 = [x_0, x_1], I_1 = [x_1, x_2], \cdots I_N = [x_N, x_{N+1}]$, we then try to approximate u by choosing our basis function (or **shape function**) $\phi = [\phi_1 \ldots \phi_N]$ that would satisfy **partition of unity**, and approximate u as a linear combination of ϕ:

$$u_h = \sum_{i=1}^{i=N} \alpha_i \phi_i \in V_h \tag{2.15}$$

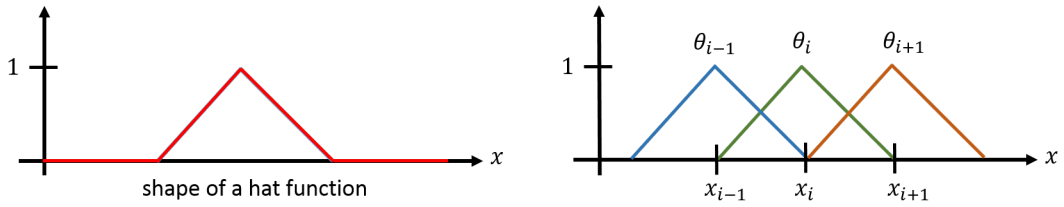


Figure 2.1: Example of basis functions: Hat functions

where $V_h = span(\theta)$. Moreover, we may also make the choice $v_h \in V_h$, that is, make u_h and v_h both as a linear combination of ϕ. We choose $v_h = \theta\bar{v}$, where $\bar{v} = [\bar{v}_1 \ldots \bar{v}_N]^T$ are completely arbitrary values at the discrete points. Substituting and rewriting equation (3) we get

$$\int u_h' v_h' - \int f v_h = \bar{v}^T \left(\int (\phi^T)'(\phi)'\alpha - \int \phi^T f \right) = 0$$

which must hold for all arbitrary values of $\bar{v}$, so

$$\underbrace{\int (\phi^T)'(\phi)'}_{S} \alpha - \underbrace{\int \phi^T f}_{r} = 0 \tag{2.16}$$

where the **stiffness matrix** S is positive definite. Since we approximate the solution by partitioning the domain into finite elements (intervals), this is called the **Finite Element Method (FEM)** and we are left with solving the standard problem $S\alpha = r$.

2.3 1D Discontinuous Galerkin Method

Suppose now we want our choice of the test function v being discontinuous on the nodes $x_1, \cdots, x_N$. Then we cannot integrate the first term $-\int u''v$ in equation (2) since v is not differentiable on the whole domain.

On each node x_j, let x_j^- represent the end of the left interval I_{j-1}, and x_j^+ represent the start of the right interval I_j. Notice that we would have our function v look like this:

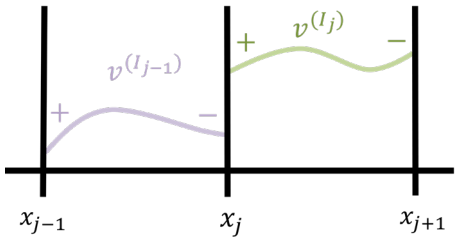


Figure 2.1.b: v being discontinuous on nodes

thus we are having two function v values, $v(x_j^-)$ from I_{j-1} and $v(x_j^+)$ from I_j, on each node x_j for $j = 1...N$. We may proceed by choosing a new basis function $\phi = [\phi_1 ... \phi_M]$ that is discontinuous at the nodes

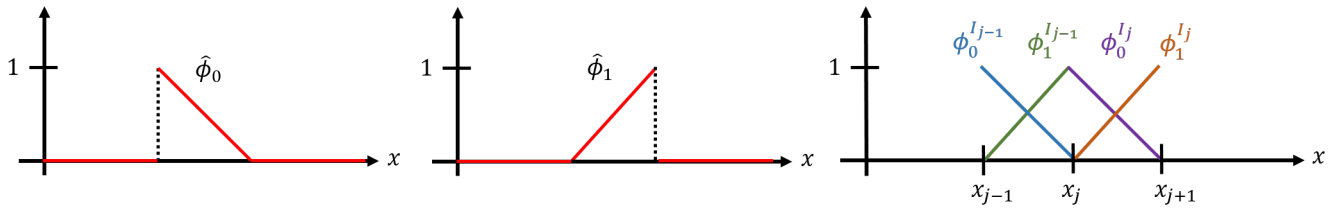


Figure 2.1.c: Example of linear basis functions, discontinuous at nodes

and make our choice of the test function v to be ϕ. So now we can do integration by part on each interval $[x_j, x_{j+1}]$ and get:

$$
\begin{aligned}
-\int u''v &= -\sum_{j=0}^{j=N} \int_{x_j}^{x_{j+1}} u''v \\
&= \sum_{j=0}^{j=N} \left(\int_{x_j}^{x_{j+1}} u'v' + u'(x_j^+)\, v(x_j^+) - u'(x_{j+1}^-)\, v(x_{j+1}^-) \right) \\
&= \sum_{j=0}^{j=N} \int_{x_j}^{x_{j+1}} u'v' + u'(x_0^+)\, v(x_0^+) + \Big(\Big(-u'(x_1^-)\, v(x_1^-) + u'(x_1^+)\, v(x_1^+) \Big) + ... \\
&\quad + \Big(-u'(x_N^-)\, v(x_N^-) + u'(x_N^+)\, v(x_N^+) \Big) \Big) - u'(x_{N+1}^-)\, v(x_{N+1}^-) \\
&= \sum_{j=0}^{j=N} \int_{x_j}^{x_{j+1}} u'v' + u'(x_0^+)\, v(x_0^+) + \sum_{j=1}^{j=N} [\boldsymbol{u'v}]_{\boldsymbol{x_j}} - u'(x_{N+1}^-)\, v(x_{N+1}^-)
\end{aligned}
\tag{2.17}
$$

of which $[\boldsymbol{u'v}]_{\boldsymbol{x_j}} = -u'(x_j^-)\, v(x_j^-) + u'(x_j^+)\, v(x_j^+)$ are called the **jumps**. Now we may define $u'(x_0^-)\, v(x_0^-) = u'(x_{N+1}^+)\, v(x_{N+1}^+) = 0$ and after adding them to the equation, we have

$$
-\int u''v = \sum_{j=0}^{j=N} \int_{x_j}^{x_{j+1}} u'v' + \sum_{j=0}^{j=N+1} [u'v]_{x_j} \tag{2.18}
$$

In order to transmit information between each jump, we define $\{\cdot\}_j$, $[\cdot]_j$ be the average and difference of the function values on x_j respectively, and rewrite each jump as

$$\begin{aligned}
[u'v]_{x_j} &= -u'(x_j^-)\,v(x_j^-) + u'(x_j^+)\,v(x_j^+) \\
&= \{\frac{1}{2}\Big(u'(x_j^+) + u'(x_j^-)\Big)\}\,[v(x_j^+) - v(x_j^-)] + [u'(x_j^+) - u'(x_j^-)]\,\{\frac{1}{2}\Big(v(x_j^+) + v(x_j^-)\Big)\} \\
&= \{u'\}_j[v]_j + [u']_j\{v\}_j
\end{aligned} \tag{2.19}$$

The second term $[u']_j\{v\}_j$ disappears since u is continuous. On the other hand, we wish to make the equation symmetric on u, v, so we add symmetric terms $\{v'\}_j[u]_j$ for each j. Together with the **penalty term**, we obtain the **weak form** with $a(u,v)$ which is a bilinear function symmetric on u, v:

$$a(u,v) \equiv \sum_{j=0}^{j=N}\int_{x_j}^{x_{j+1}} u'v' + \sum_{j=0}^{j=N+1}\Big(\{u'\}_j[v]_j + \underbrace{\{v'\}_j[u]_j}_{\text{symmetric term}}\Big) + \underbrace{\gamma\sum_{j=0}^{N+1}\frac{1}{|I_j|}[u]_j[v]_j}_{\text{penalty term}} \tag{2.20}$$

(Notice that when the test function v is continuous, the weak form is essentially the same as the one in FEM [1]).

Assume u is to be approximated by a linear combination of ϕ

$$u_h = \sum_{j=1}^{j=M}\alpha_j\phi_j \tag{2.21}$$

so we have

$$\begin{aligned}
a(u_h, v_h) &= \int f\,v_h + \text{symmetric term} + \text{penalty term} \\
a\Big(\sum_{j=1}^{j=M}\alpha_j\phi_j, \phi_i\Big) &= \int f\,\phi_i + \text{symmetric term} + \text{penalty term} \qquad \text{for } i = 1...M \\
\sum_{j=1}^{j=M}\underbrace{a(\phi_j,\phi_i)}_{S_{ij}}\alpha_j &= \underbrace{\int f\,\phi_i}_{r_i} + \text{symmetric term} + \text{penalty term} \qquad \text{for } i = 1...M
\end{aligned} \tag{2.22}$$

Again we are left with the standard problem $S\alpha = r$. This is called the **Discontinuous Galerkin Finite Element Method (DG-FEM)**.

[1] since the second term and the penalty term would disappear.

2.4 Essence of the Penalty Term

Suppose we have

$$\sum_{I_j}\int_{x_j}^{x_{j+1}} u'v' + \sum_{I_j}\Big(\{u'\}_j[v]_j + \{v'\}_j[u]_j\Big) = \int fv$$

and we wish to solve for u. Writing in matrix form, it is easy to see that the stiffness matrix in LHS may be singular [2]. Therefore we wish to add a **penalty term** to LHS such that the matrix is always invertible and the solution can be uniquely determined. Therefore we define

$$a(u,v) \equiv \sum_{I_j}\int_{x_j}^{x_{j+1}} u'v' + \sum_{I_j}\Big(\{u'\}_j[v]_j + \{v'\}_j[u]_j\Big) + \underbrace{\gamma\sum_{I_j}\frac{1}{|I_j|}[u]_j[v]_j}_{\text{penalty term}} \tag{2.23}$$

and we claim that the stiffness matrix S defined by $a(u,v)$ is positive definite. We are going to show that, for a chosen A, if γ is large enough we would always have

$$a(u,v) \geq A||u||_{1,h}{}^2 \quad \forall u \in V_h \tag{2.24}$$

where V_h is the finite element(DG) space and $||u||_{1,h}$ is a well-defined norm. Then we know that S is positive definite and thus invertible, and we are done.

With the conventional norm $||\cdot||_I = (\cdot,\cdot)_I{}^{\frac{1}{2}}$ in which the inner product is defined as $(f,g)_I = \int_I fg$ for any functions f, g, we can now define $||u||_{1,h}$:

Definition 2.2.1:

$$||u||_{1,h} = \Big(\sum_{I_j}||u'||_{I_j}{}^2 + \sum_{I_j}|\{u'\}_j|^2|I_j| + \sum_{I_j}\frac{\gamma}{|I_j|}|[u]_j{}^2\Big)^{\frac{1}{2}}$$

and $||u||_{1,h}$ is a well-defined norm. In particular, it satisfies the norm property

$$||u||_{1,h} \geq 0 \quad and \quad ||u||_{1,h} = 0 \iff u = 0$$

Proof: " $\Leftarrow$ " is trivial.
" $\Rightarrow$ " : We must have both $||u'||_{I_j}{}^2$ and $|\{u'\}_j|^2$ equal to zero. The former term equals to zero implies that u is a piecewisely constant function . The latter term equals to zero implies that there are no differences between the function values of u on all nodes including the boundary, whose value is defined to be zero. Hence we must have u being a constant zero function.

Before proceeding we would need the following:

Theorem 2.2.2: (*Trace Inequality*) Let v be a function on domain Ω and H be the length of Ω.
Then we have

$$\int_{\partial\Omega}|v|^2 ds \leq cH^{-1}||v||_\Omega{}^2 + cH||\nabla||_\Omega{}^2$$

for some constant c. Or in 1D sense,

$$|v(a)|^2 + |v(b)|^2 \leq cH^{-1}||v||_I{}^2 + cH||v'||_I{}^2 \tag{2.25}$$

with $H = |I|$ being the length of the interval.

[2] Take u to be a non-zero constant function and LHS would become zero.

Theorem 2.2.3: (*Inverse Inequality*) Let u be a function on the interval I. Then we have

$$||u'||^2 \leq c|I|^{-2}||u||_I{}^2$$

Or

$$|I|^2||u'||^2 \leq c||u||_I{}^2 \tag{2.26}$$

for some constant c.

Claim 2.2.4: We have

$$\sum_{I_j}|I_j|\,|\{u'\}_j|^2 \leq c\sum_{I_j}||u'||_{I_j}{}^2$$

for some constant c.

Proof: Applying *Trace Inequality* and *Inverse Inequality*, we have

$$\begin{aligned}
&\sum_{I_j}|I_j|\,|\{u'\}_j|^2 \\
(\textit{Trace.}, \text{equation (14)}) \leq\ &\sum_{I_j}|I_j|\Big(\hat{c}\,|I_j{}^{-1}|||u'||_{I_j}{}^2+\hat{c}\,|I_j|\,||u''||_{I_j}{}^2\Big) \\
=\ &\hat{c}\sum_{I_j}\Big(|I_j||I_j|^{-1}||u'||_{I_j}{}^2+|I_j|^2||u''||_{I_j}{}^2\Big) \\
=\ &\hat{c}\sum_{I_j}\Big(||u'||_{I_j}{}^2+|I_j|^2||u''||_{I_j}{}^2\Big) \\
(\textit{Inverse.}, \text{equation (15)}) \leq\ &\hat{c}\sum_{I_j}\Big(||u'||_{I_j}{}^2+\check{c}||u'||_{I_j}{}^2\Big) \\
=\ &c\sum_{I_j}||u'||_{I_j}{}^2
\end{aligned}$$

for some constant c.

Claim 2.2.5: We have

$$\sum_{I_j}\{u'\}_j\,[u]_j \leq c_1\,\delta\sum_{I_j}||u'||_{I_j}{}^2+\frac{c_2}{\delta}\sum_{I_j}\frac{1}{I_j}|\,[u]_j\,|^2$$

for some constant c_1, c_2.

Proof:

$$\begin{aligned}
&\sum_{I_j}\{u'\}_j\,[u]_j \\
\leq\ &\sum_{I_j}|\{u'\}_j||[u]_j| \\
(\textit{AM-GM.}) \leq\ &\sum_{I_j}\Big(\frac{\delta}{2}|I_j||\{u'\}_j|^2+\frac{1}{\delta|I_j|}[u]_j{}^2\Big) \\
(\textit{Claim 2.2.4}) \leq\ &\sum_{I_j}c_1\,\delta\,||u'||_{I_j}{}^2+\sum_{I_j}\frac{1}{\delta|I_j|}[u]_j{}^2 \\
=\ &c_1\,\delta\sum_{I_j}||u'||_{I_j}{}^2+\frac{c_2}{\delta}\sum\frac{1}{|I_j|}|u|_{I_j}{}^2
\end{aligned}$$

for arbitrary constant δ and some constant c_1, c_2.

Finally cleaning up:

$$a(u,u) - A||u||_{1,h}{}^2$$

$$\begin{aligned}
\text{(equation (12), } \textit{Def. 2.2.1}) = \quad & \Big(\sum_{I_j} ||u'||_{I_j}{}^2 - 2\sum_{I_j}\{u'\}_j[u']_j + \sum_{I_j}\frac{\gamma}{|I_j|}|[u]_j|^2\Big) \\
& - A\Big(\sum_{I_j} ||u'||_{I_j}{}^2 + \sum_{I_j}|I_j||\{u'\}_j|^2 + \sum_{I_j}\frac{\gamma}{|I_j|}|[u]_j|^2\Big) \\
= \quad & (1-A)\sum_{I_j} ||u'||_{I_j}{}^2 + (1-A)\sum_{I_j}\frac{\gamma}{I_j}|[u]_j|^2 \\
& - 2\sum_{I_j}\{u'\}_j[u'] - A\sum_{I_j}|I_j||\{u'\}_j|^2 \\
(\textit{Claim 2.2.4, 2.2.5}) \geq \quad & (1-A)\sum_{I_j} ||u'||_{I_j}{}^2 + (1-A)\sum_{I_j}\frac{\gamma}{I_j}|[u]_j|^2 \\
& - \Big(c_1\delta\sum_{I_j} ||u'||_{I_j}{}^2 + \frac{c_2}{\delta}\sum\frac{1}{|I_j|}|u|_{I_j}{}^2\Big) - c_3 A\sum_{I_j} ||u'||_{I_j}{}^2 \\
= \quad & (1-A-c_1\delta-c_3A)\sum_{I_j} ||u'||_{I_j}{}^2 + (\gamma-\gamma A-\frac{c_2}{\delta})\sum_{I_j}\frac{1}{|I_j|}|[u]_j|^2 \\
\geq \quad & 0
\end{aligned}$$

for γ large enough[3] according to the chosen A and δ, which is equivalent to equation (13). *Q.E.D.*

[3] For instant, choose $A = 1/(2c_3)$ and $\delta = -1/(2c_1c_3)$, then take $\gamma \geq (4c_1c_2c_3{}^2)/(1-2c_3)$.

2.5 General Discountinous Galerkin Method (DG-FEM)

Consider the general problem:

$$\begin{cases} -\triangle u &= f & \text{in } \Omega \\ u &= g_D & \text{on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} &= g_N & \text{on } \Gamma_N \end{cases} \tag{2.27}$$

which is the **Poisson's Equation** $-\triangle u = f$, where f is given and u is the unknown function on a domain Ω, together with the given g_D on boundary Γ_D (**Dirichlet boundary condition**) and the given g_N on boundary Γ_N (**Neumann boundary condition**).The domain Ω can be of different dimension. The following figure shows an example of a 2D domain.

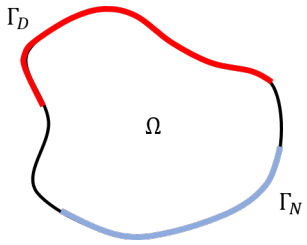


Figure 2.2: Example of a 2D domain

Multiply each side by test function v and using integration by part, we have

$$\begin{aligned} -\int_\Omega \triangle u\, v\, dx &= \int_\Omega \nabla u \cdot \nabla v\, dx - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n})\, v\, ds \\ &= \int_\Omega \nabla u \cdot \nabla v\, dx - \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v\, ds \\ &= \int_\Omega f v\, dx \end{aligned} \tag{2.28}$$

in which we have reduced the equation from the strong form to a more desirable weak form.

Take 2D problem as an example of the general case. We would like to approximate u by a linear combination of some chosen basis functions in a finite element space. We may consider partitioning the domain into finite elements in the form of triangles:

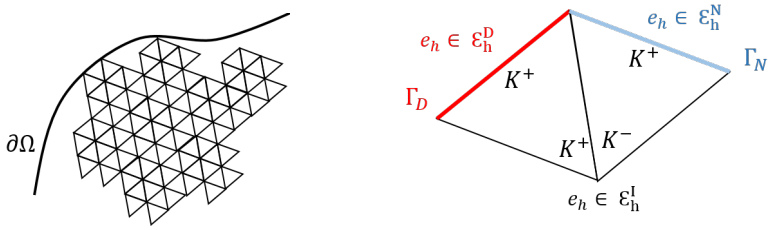


Figure 2.3.b: Partition of a 2D domain in triangles and edge sets: $\mathcal{E}_h^I$, $\mathcal{E}_h^D$ and $\mathcal{E}_h^N$

Denote a triangle by K and its diameter by h, we let $\mathcal{T}_h$ be the set of all triangles and define the following sets:

$$\begin{cases} \mathcal{E}_h^I &= \{e_h : e_h = \partial K^+ \cap \partial K^- \quad \forall K \in \mathcal{T}_h\}\text{: set of } \textbf{interior edges} \\ \mathcal{E}_h^D &= \{e_h : e_h = \partial K^+ \cap \Gamma_D \quad \forall K \in \mathcal{T}_h\}\text{: set of } \textbf{boundary edges on } \Gamma_D \\ \mathcal{E}_h^N &= \{e_h : e_h = \partial K^+ \cap \Gamma_N \quad \forall K \in \mathcal{T}_h\}\text{: set of } \textbf{boundary edges on } \Gamma_N \\ \mathcal{E}_h &= \mathcal{E}_h^I \cup \mathcal{E}_h^D. \end{cases} \tag{2.29}$$

(Note that the signs of the triangles are declared relatively; and interior edges must align with triangles of different signs. Triangles at the boundary are always denoted by K^+.) Thus now we can make our choice of basis functions to be discontinuous across the edges, and proceed on DG-FEM. Take v to be the chosen basis functions, we then have

$$\begin{aligned}
-\int_\Omega \triangle u\, v\, dx &= -\sum_{K\in\mathcal{T}_h}\int_K \triangle u\, v\, dx \\
&= \sum_{K\in\mathcal{T}_h}\int_K \nabla u\cdot\nabla v\, dx - \sum_{K\in\mathcal{T}_h}\int_{\partial K}\frac{\partial u}{\partial \mathbf{n}} v\, ds \\
&= \sum_{K\in\mathcal{T}_h}\int_K \nabla u\cdot\nabla v\, dx - \sum_{e_h\in\mathcal{E}_h^D}\int_{e_h}\frac{\partial u}{\partial \mathbf{n}} v\, ds - \sum_{e_h\in\mathcal{E}_h^N}\int_{e_h}\frac{\partial u}{\partial \mathbf{n}} v\, ds \\
&\qquad - \sum_{e_h\in\mathcal{E}_h^I}\int_{e_h}\Big(\frac{\partial u^+}{\partial \mathbf{n} n^+}v^+ + \frac{\partial u^-}{\partial \mathbf{n} n^-}v^-\Big) ds \\
&= \int_\Omega f\, v\, dx
\end{aligned} \tag{2.30}$$

We wish to find out a symmetric bilinear function $a_h(u,v)$, whose stiffness matrix is positive definite, to solve for the unknown u. For simplicity, we denote

$$\int_k fg\, dx \equiv (f,g)_K \qquad \text{and} \qquad \int_{e_h} fg\, ds \equiv <f,g>_{e_h}$$

Also we define the notations $\{\cdot\}$, $[\cdot]$ on function and norm derivative:

$$\begin{aligned}
\{u\} &\equiv \frac{1}{2}(u^+ + u^-) \qquad \text{and} \qquad [u] \equiv u^+ - u^- \\
\{\partial_n u\} &\equiv \frac{1}{2}\Big(\frac{\partial u^+}{\partial \mathbf{n}^+} + \frac{\partial u^-}{\partial \mathbf{n}^+}\Big) \qquad \text{and} \qquad [\partial_n u] \equiv \frac{\partial u^+}{\partial \mathbf{n}^+} - \frac{\partial u^-}{\partial \mathbf{n}^+}
\end{aligned} \tag{2.31}$$

then look into the term regarding the set of interior edges. By noticing the direction of the norm derivatives and the continuity of u, we have

$$\begin{aligned}
\sum_{e_h\in\mathcal{E}_h^I}\int_{e_h}\Big(\frac{\partial u^+}{\partial \mathbf{n}^+}v^+ + \frac{\partial u^-}{\partial \mathbf{n}^-}v^-\Big) ds &= \sum_{e_h\in\mathcal{E}_h^I}\int_{e_h}\Big(\frac{\partial u^+}{\partial \mathbf{n}^+}v^+ - \frac{\partial u^-}{\partial \mathbf{n}^+}v^-\Big) ds \\
&= \sum_{e_h\in\mathcal{E}_h^I}\Big(<\frac{\partial u^+}{\partial \mathbf{n}^+}, v^+>_{e_h} - <\frac{\partial u^-}{\partial \mathbf{n}^+}, v^->_{e_h}\Big) \\
\text{(after rearranging)} &= \sum_{e_h\in\mathcal{E}_h^I}\Big(<\{\partial_n u\}, [v]>_{e_h} + <[\partial_n u], \{v\}>_{e_h}\Big) \\
\text{(since } [\partial_n u]=0) &= \sum_{e_h\in\mathcal{E}_h^I}<\{\partial_n u\}, [v]>_{e_h}
\end{aligned} \tag{2.32}$$

Rewrite equation (19) with the notations defined, we have :

$$\begin{aligned}
-(\triangle u, v) &= \sum_{K\in\mathcal{T}_h}(\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h^I}<\{\partial_n u\}, [v]>_{e_h} - \sum_{e_h\in\mathcal{E}_h^D}<\partial_n u, v>_{e_h} \\
&\qquad - \sum_{e_h\in\mathcal{E}_h^N}<\partial_n u, v>_{e_h} \\
&= (f,v)
\end{aligned} \tag{2.33}$$

To make the expression in the left hand side symmetric on u, v, we add the artificial terms

$$-\sum_{e_h\in\mathcal{E}_h^I}<\{\partial_n v\}, [u]>_e \qquad \text{and} \qquad -\sum_{e_h\in\mathcal{E}_h^D}<\partial_n v, u>_{e_h}$$

Notice that the first term equals to 0 since $[u]=0$ by continuity of u. In the second term, $u=g_D$ which is given, so we add this term to both sides of the equation. Also $\sum_{e_h\in\mathscr{E}_h^N}<\partial_n u,v>_{e_h}=\sum_{e_h\in\mathscr{E}_h^N}<g_N,v>$ is known, so this term is moved to the right hand side.

To make the bilinear form coercive (positive definite), we add the penalty terms

$$\sum_{e_h\in\mathscr{E}_h^I}\frac{\gamma}{|e_h|}<[u],[v]>_{e_h}=0 \quad \text{and} \quad \sum_{e_h\in\mathscr{E}_h^D}\frac{\gamma}{|e_h|}<u,v>_{e_h}=\sum_{e_h\in\mathscr{E}_h^D}\frac{\gamma}{|e_h|}<g_D,v>_{e_h}.$$

The first and second terms are added to the left hand side, while the third term is added to the right hand side. With the unknown values stay in the left hand side and away from the right hand side, we may define the left hand side as a bilinear form $a_h(u,v)$:

$$\begin{aligned} a_h(u,v)\equiv \quad & \sum_{K\in\mathscr{T}_h}(\nabla u,\nabla v)_K-\sum_{e_h\in\mathscr{E}_h^I}\left(<\{\partial_n u\},[v]>_{e_h}+<\{\partial_n v\},[u]>_{e_h}-\frac{\gamma}{|e_h|}<[u],[v]>_{e_h}\right) \\ & -\sum_{e_h\in\mathscr{E}_h^D}\left(<\partial_n u,v>_{e_h}+<\partial_n v,u>_{e_h}-\frac{\gamma}{|e_h|}<u,v>_{e_h}\right) \end{aligned} \tag{2.34}$$

Note that the penalty terms are added to the interior and Dirichlet edges but not to the Neumann edges. This is because the Neumann boundary terms have been moved to the right hand side.

Also, adopting the notation that on the boundary edges

$$\{\partial_n u\}=\partial_n u, \quad \{\partial_n v\}=\partial_n v, \quad [u]=u, \quad [v]=v,$$

then we can write the bilinear form in equation (2.25) as a more compact form (recall that $\mathscr{E}_h=\mathscr{E}_h^I\cup\mathscr{E}_h^D$)

$$a_h(u,v)\equiv \sum_{K\in\mathscr{T}_h}(\nabla u,\nabla v)_K-\sum_{e_h\in\mathscr{E}_h}\left(<\{\partial_n u\},[v]>_{e_h}+<\{\partial_n v\},[u]>_{e_h}-\frac{\gamma}{|e_h|}<[u],[v]>_{e_h}\right) \tag{2.35}$$

We also define the right hand side, which depends only on v and is composited of known values, as a function $F(v)$ by

$$F(v)\equiv(f,v)+\sum_{e_h\in\mathscr{E}_h^D}\left(\frac{\gamma}{|e_h|}<g_D,v>_{e_h}-<g_D,\partial_n v>_{e_h}\right)+\sum_{e_h\in\mathscr{E}_h^N}<g_N,v>_{e_h} \tag{2.36}$$

3 CONSTRUCTION OF THE STIFFNESS MATRIX

Overview: In this section, the main idea and technics used in the construction of the stiffness matrix will be introduced.

3.1 MAIN IDEA OF THE CONSTRUCTION

To start, let's assume there is a domain Ω, whose boundary $\partial\Omega$ consists of Γ_D and Γ_N, and look at the following problem.

$$\begin{cases} -\triangle u &= f & \text{in } \Omega \\ u &= g_D & \text{on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} &= g_N & \text{on } \Gamma_N \end{cases} \tag{3.1}$$

To approximate the function value u, we need to choose some points in this domain as our nodal

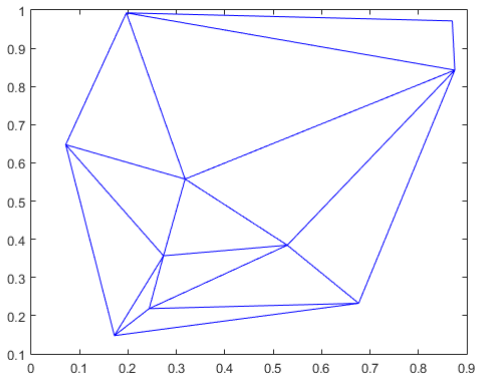


Figure 3.1: A random 2D mesh

points, which we can calculate their exact values and then use them to produce a interpolotation. We denote the set of such points by $N = \{N_1, N_2, ...\}$. Then we can devide the domain into several elements, according to the points we choose. In 2D case, we usually divided the domain into triangles or rectangles; in 3D case, we usually divided the domain into tetrahedrons or cubes. In figure 3.1, we devide a 2D domain into a combination of triangles. After that, a set of basic functions, corresponding to the point we choose, should be set to do the approximation. That is, whatever those nodal points' value are, we can use the set of basic functions to form a linear combination, which can exactly fit in those values. In this way, the set of functions can be used to represent the value of every point we choose.

However, since the mesh varies from one to another, finding the basic functions and doing the subsequent calculation is quite hard to be finished directly in the original mesh. As a result, we choose to set a master cell (or master element), whose shape is relatively 'beautiful' and thus it's easy to find the basic functions for it. Then for every calculation we need to do in the elements in mesh, we can do the same operation in master cell first, since all the information we need is already known or easy to get for the master cell. As for the calculation in elements, we will use maps, which connect the elements and the master cell, to transform the result get from the master cell to what we want. The map from the master cell to each element in the mesh, while different from element to element, can be set easily, since we know the coordinate of all the nodes in the mesh. Following such steps, all the calculation can be finished in the master cell and will be much simpler. All we need to do is to store

the template, which contains the value calculated in the master cell, and find out the map from each element to the master cell.

3.2 THE MASTER CELL AND THE BASIC FUNCTION

As the first step, we need to set our master cell. For convenience and simplicity, we always choose the origin and those points which are 1 unit away from the origin on the axis to form the standard master cell. The following figures shows the master cell we use for 2D case, which is a triangle, and 3D case, which is a tetrahedron.

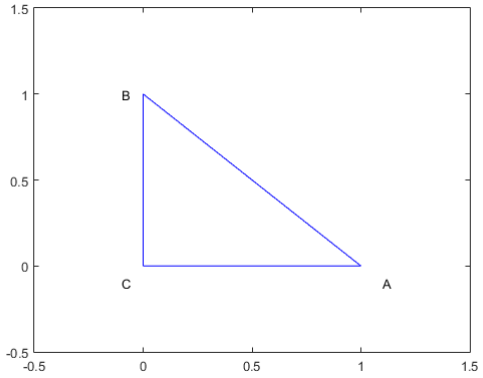


Figure 3.2: master cell for 2D: a triangular

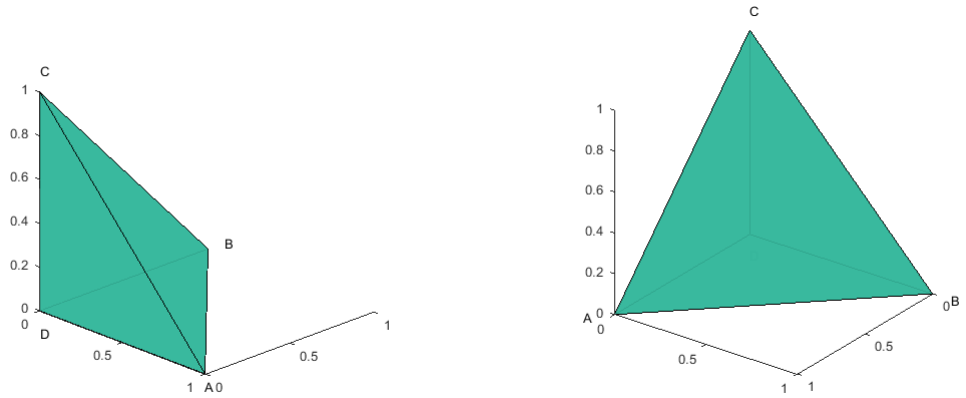


Figure 3.3: master cell for 3D: a tetrahedron

There are two main advantages of setting in this way. Firstly, the basic functions of the master cell will be easy to find, since its shape is quite 'beautiful'. Secondly, all the subsequent integrations, especially the integrations on the edge or face, will be respectively easy to calculate.

How to find the basic functions for the master cell? Basically, we use lagrange interplotation method. Take 2D case as an example. There are several steps we can follow:

step 1 decide degree of our basic functions

step 2 decide the degree of freedom

step 3 choose the set of nodal points

step 4 find the set of basic functions for master cell

The first step we take is to decide what degree our basic functions are. In general, the higher degree of basic functions is, the higher the accuracy is. At the same time, the complexity of calculation will also be higher. In a word, the accuracy of the approximation you want and the cost of calculation you can afford determine the degree of the basic funcifons you use.

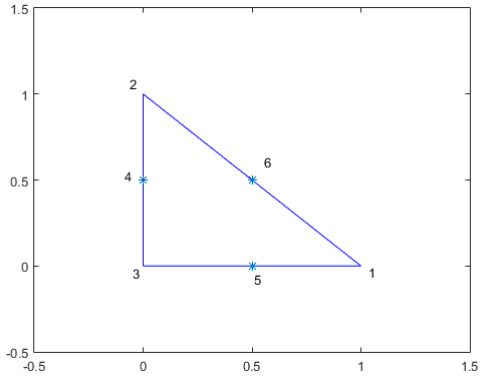


Figure 3.4: Point we choose for quadratic basic functions

The second step is to decide the degree of freedom. It is followed by the first step. If you choose linear functions, then we need two points on each eage to make the function exact. So the degree of freedom is 3. If you choose quadratic functions, then we need three points on each eage to make the function exact. So the degree of freedom is 6.

Then we need to choose the nodal points. For convenience, we usually use those special points. When the degree of freedom is 3, we choose 3 vertex of the triangle as our nodal points. When the degree of freedom is 6, we will add the midpoint of each edge to our nodal point set.

Finally, we can decide the basic functions. By Lagrange interpolatation mathod, each nodal point N_i in the master cell has a corresponding basic function ϕ_i so that $\phi_j(N_i) = 1$ when $j = i$ and $\phi_j(N_i) = 0$ when $j \neq i$. Hence, the number of basic functions always equal to the degree of freedom.

For example, if we use linear basic funcion, then the degree of freedom is 3.

Let k = master triangle

$P_k = p_1(k) = span\{1, x, y\}$

The we have basic functions: $\phi_1 = x \quad \phi_2 = y \quad \phi_3 = 1 - x - y$

where ϕ_1, ϕ_2, ϕ_3 are corresponding function to point A, B, C respectively.(in Figure3.2) If we use quadratic basic functions, then the degree of freedom becomes 6.

Let k = master triangle

$P_k = p_2(k) = span\{1, x, y, x^2, y^2, xy\}$

The we have basic functions:

$\phi_1 = 2x(0.5 - x) \quad \phi_2 = 2y(0.5 - y) \quad \phi_3 = 2(1 - x - y)(0.5 - x - y)$

$\phi_4 = 4y(1 - x - y) \quad \phi_5 = 4x(1 - x - y) \quad \phi_6 = 4xy$

3.3 THE MAP FROM MASTER CELL TO ELEMENTS IN MESH

After set down the master cell, we need to calculate the map from master cell to mesh element. Note that the information we need now from the map is only how each nodal point in elements map to the master cell. For example, suppose the nodal points of the master cell is $[A,B,C]$ for 2D case, and we have a master element whose nodal points are$[c,b,a]$, what we only need to know is how $[A,B,C]$ match $[c,b,a]$. For convenience of programmming, we usually map them in order. That is, we map c to A, b to B, a to C. Then we have the map already.

Then we need to calculate some other information about the map. Here we use isomorphic map in 2D case for example, and 3D case is exactly the same. Suppose there is a local node points (x_j, y_j), $j = 1,2,...,M$ in mesh element K. Also, suppose we have a nodal point $(\hat{x}_j, \hat{y}_j)$, $j = 1,2,...,M$ in master cell and corresponding basic functions $\hat{\phi}_j(\hat{x}_j, \hat{y}_j)$, $j = 1,2,...,M$. Note that M is the value of the degree of freedom. Then for every point (x, y) in K, we have the following map $\mathbf{T_e}$:

$$\begin{cases} x = \sum_{j=1}^{M} x_j \hat{\phi}_j(\hat{x}_j, \hat{y}_j) \\ y = \sum_{j=1}^{M} y_j \hat{\phi}_j(\hat{x}_j, \hat{y}_j) \end{cases} \tag{3.2}$$

Since $\hat{\phi}_j(\hat{x}_j, \hat{y}_j)$ is known for $1 \le j \le M$, so we can easily write out the expression of $\hat{x}$ and $\hat{y}$ with x and y:

$$\begin{pmatrix} x \\ y \end{pmatrix} = A \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \vec{b} \tag{3.3}$$

Through this formula we can compute some other things, like jacobian of A and A', which will be used in the later computation.

Now we use an example to make things clearer. Suppose we have the same master triangle $\hat{k}$ as what is shown in figure 3.2 and another triangle element k. We set the basic functions for $\hat{k}$ is like the following:

$$\begin{cases} \phi_A = \hat{x} \\ \phi_B = \hat{y} \\ \phi_C = 1 - \hat{x} - \hat{y} \end{cases} \tag{3.4}$$

Then for any point (x, y) in triangle k, we have

$$\begin{cases} x = x_a \phi_A + x_b \phi_B + x_c \phi_C \\ y = y_a \phi_A + y_b \phi_B + y_c \phi_C \end{cases} \tag{3.5}$$

After simplification, we will get

$$\begin{cases} x = (x_a - x_c)\hat{x} + (x_b - x_c)\hat{y} + x_c \\ y = (y_a - y_c)\hat{x} + (y_b - y_c)\hat{y} + y_c \end{cases} \tag{3.6}$$

after transformation, the equation becomes:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_a - x_c & x_b - x_c \\ y_a - y_c & y_b - y_c \end{pmatrix} * \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \begin{pmatrix} x_c \\ y_c \end{pmatrix} \tag{3.7}$$

So we have

$$\begin{pmatrix} x_a - x_c & x_b - x_c \\ y_a - y_c & y_b - y_c \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{pmatrix} \tag{3.8}$$

and the jacobian

$$J_{(\hat{x},\hat{y})} = det \begin{pmatrix} x_a - x_c & x_b - x_c \\ y_a - y_c & y_b - y_c \end{pmatrix} \tag{3.9}$$

Then we can write out the inverse formulation

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \frac{1}{|J_{(\hat{x},\hat{y})}|} \begin{pmatrix} y_a - y_c & x_c - x_b \\ y_c - y_a & x_b - x_c \end{pmatrix} * \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \tag{3.10}$$

Then we can get

$$\frac{1}{|J_{(\hat{x},\hat{y})}|} \begin{pmatrix} x_a - x_c & x_b - x_c \\ y_a - y_c & y_b - y_c \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{pmatrix} \tag{3.11}$$

Then we have the information we need in later integration.

Notice that the above map can can be written in a more general form:

$$\vec{x} = \begin{pmatrix} \vec{x_1} - \vec{x_0} & \vec{x_2} - \vec{x_0} & \dots & \vec{x_M} - \vec{x_0} \end{pmatrix} * \vec{\hat{x}} + \vec{x_0} \tag{3.12}$$

which can be expanded to higher dimensional cases.

3.4 ELEMENT-ELEMENT TERM

Now we look back to the general equation $a_h(u,v) = F(v)$ and try to construct the matrix according to this eqation. Before we start, we will give a general look at the problem. The aim is to build the LHS stiffness matrix A. The matrix A is a combination of several local blocks. In general, the size of one local block is $M \times M$, where M is the degree of freedom of each elements. Also, there will be $K \times K$ local blocks in the global matrix, where K is the number of elements. Hence the size of the glbal matrix is $KM \times KM$.

The local block is the basic unit in our calculation. The diagonal local blocks, for instance, the (i,i) block, describe the integration inside the i-th elements. The non-diagonal local blocks, for instance, the (i,j) and (j,i) block, describe the integration on the common edge (or face) between the i-th and j-th elements. Hence we will find the final global matrix is sparse, since (i,j) block will be zero block when $i \neq j$ and the i-th and j-th element have no common edge (or face).

We would like to use the notation $a(\phi_j^{(k_1)}, \phi_i^{(k_2)})$to represent the (i,j) element of (k_2,k_1) block. Recall that

$$a_h(u,v) \equiv \sum_{K\in\mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h\in\mathcal{E}_h} \left(< \{\partial_n u\}, [v] >_{e_h} + < \{\partial_n v\}, [u] >_{e_h} - \frac{\gamma}{|e_h|} < [u], [v] >_{e_h} \right)$$

We can devide it into three parts:

1. range over cell
2. range over interior edges
3. range over boundary (Dirichilet) edges

In our practical compuatation, we calculate these three parts separately. In this section we will discuss the first part, the integration inside the element, which we called it element-element term.

As mentioned before, we need to map the function integrated into the basic functions of master element. When for 2D case, it goes like this:

$$\begin{aligned}
\int_k \nabla u \nabla v ds &= \int\int (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}) dxdy \\
&= \int\int (\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}) dxdy \\
&= \int\int [(\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y})] |J_{(\hat{x},\hat{y})}| d\hat{x}d\hat{y} \\
&= \int\int F(\hat{x},\hat{y}) d\hat{x}d\hat{y} \\
\text{(Gaussian quadrature)} &= \sum w_i F(\hat{x}_i, \hat{y}_i)
\end{aligned}$$

And for 3D case, which is almost the same, we have:

$$
\begin{aligned}
\int_k \nabla u \nabla v ds &= \int\int (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z})(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial u}{\partial z}) dxdydz \\
&= \int\int (\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} + \frac{\partial u}{\partial z}\frac{\partial v}{\partial z}) dxdydz \\
&= \int\int [(\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial x})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial x}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial y})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial y}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial z} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial z} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z})]|J_{(\hat{x},\hat{y},\hat{z})}| d\hat{x}d\hat{y}d\hat{z} \\
&= \int\int F(\hat{x},\hat{y},\hat{z}) d\hat{x}d\hat{y}d\hat{z} \\
\text{(Gaussian quadrature)} &= \sum w_i F(\hat{x}_i, \hat{y}_i, \hat{z}_i)
\end{aligned}
$$

where $|J_{(\hat{x},\hat{y})}|$ and $|J_{(\hat{x},\hat{y},\hat{z})}|$is the jacobian of the transformation map, which is introduce in Section 3 Also, since we have the map, the value of these terms like $\frac{\partial \hat{x}}{\partial x}$, $\frac{\partial \hat{x}}{\partial y}$, $\frac{\partial \hat{x}}{\partial z}$, $\frac{\partial \hat{y}}{\partial x}$, $\frac{\partial \hat{y}}{\partial y}$ and $\frac{\partial \hat{y}}{\partial z}$ are known, which we have introduce the method to calculate them in Section 3 as well. So we can calculate the element-element term directly.

Since this kind of integration is integration over the cell, so all the information it need is those basic functions of that cell. So the blocks for element item can only appear in diagonal blocks. $\sum_k (\nabla \phi_j^{k_1}, \nabla \phi_i^{k_2})$ will only be nonzero when $k_1 = k_2 = T$, which is a part of the (i,j) element of (T, T) block.

3.5 Interior term

Now we move to the second cases, that is, integration on interior edges. This kind of case is much more complicate to calculate than the element item. We now list the formula for them first.

Fix $e \in \mathscr{E}_h^I$, there is only four blocks are nonzeros

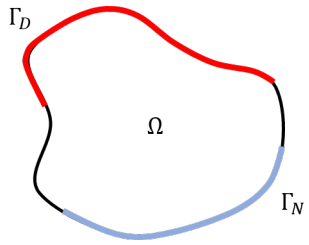


Figure 3.5: An interior edge between two elements

- $a(\phi_j^{k^+}, \phi_i^{k^+})$: the (i,j) element of (k^+, k^+) block.
- $a(\phi_j^{k^-}, \phi_i^{k^-})$: the (i,j) element of (k^-, k^-) block.
- $a(\phi_j^{k^+}, \phi_i^{k^-})$: the (i,j) element of (k^-, k^+) block
- $a(\phi_j^{k^-}, \phi_i^{k^+})$: the (i,j) element of (k^+, k^-) block

Use $I(\phi_j^{k1}, \phi_i^{k2})$ to denote the interior jump term between $k1$ and $k2$. Then

$$\begin{aligned} I(\phi_j^{k^+}, \phi_i^{k^+}) = &-\sum_e \Big(\frac{1}{2} < \partial_n(\phi_j^{k^+})^+ + \partial_n(\phi_j^{k^+})^-), (\phi_i^{k^+})^+ - (\phi_i^{k^+})^- >_e \\ &+ \frac{1}{2} < \partial_n(\phi_i^{k^+})^+ + \partial_n(\phi_i^{k^+})^-), (\phi_j^{k^+})^+ - (\phi_j^{k^+})^- >_e \\ &- \frac{\gamma}{|e_n|} < (\phi_j^{k^+})^+ - (\phi_j^{k^+})^-, (\phi_i^{k^+})^+ - (\phi_i^{k^+})^- >_e \Big) \\ = &-\frac{1}{2} < \partial_n(\phi_j^{k^+})^+, (\phi_i^{k^+})^+ >_e - \frac{1}{2} < \partial_n(\phi_i^{k^+})^+, (\phi_j^{k^+})^+ >_e + \frac{\gamma}{|e_n|} < (\phi_j^{k^+})^+, (\phi_i^{k^+})^+ >_e \end{aligned} \tag{3.13}$$

$$\begin{aligned} I(\phi_j^{k^-}, \phi_i^{k^-}) = &-\sum_e \Big(\frac{1}{2} < \partial_n(\phi_j^{k^-})^+ + \partial_n(\phi_j^{k^-})^-), (\phi_i^{k^-})^+ - (\phi_i^{k^-})^- >_e \\ &+ \frac{1}{2} < \partial_n(\phi_i^{k^-})^+ + \partial_n(\phi_i^{k^-})^-), (\phi_j^{k^-})^+ - (\phi_j^{k^-})^- >_e \\ &- \frac{\gamma}{|e_n|} < (\phi_j^{k^-})^+ - (\phi_j^{k^-})^-, (\phi_i^{k^-})^+ - (\phi_i^{k^-})^- >_e \Big) \\ = &+\frac{1}{2} < \partial_n(\phi_j^{k^-})^-, (\phi_i^{k^-})^- >_e + \frac{1}{2} < \partial_n(\phi_i^{k^-})^-, (\phi_j^{k^-})^- >_e + \frac{\gamma}{|e_n|} < (\phi_j^{k^-})^-, (\phi_i^{k^-})^- >_e \end{aligned} \tag{3.14}$$

$$\begin{aligned} I(\phi_j^{k^+}, \phi_i^{k^-}) = &-\sum_e \Big(\frac{1}{2} < \partial_n(\phi_j^{k^+})^+ + \partial_n(\phi_j^{k^+})^-), (\phi_i^{k^-})^+ - (\phi_i^{k^-})^- >_e \\ &+ \frac{1}{2} < \partial_n(\phi_i^{k^-})^+ + \partial_n(\phi_i^{k^-})^-), (\phi_j^{k^+})^+ - (\phi_j^{k^+})^- >_e \\ &- \frac{\gamma}{|e_n|} < (\phi_j^{k^+})^+ - (\phi_j^{k^+})^-, (\phi_i^{k^-})^+ - (\phi_i^{k^-})^- >_e \Big) \\ = &-\frac{1}{2} < \partial_n(\phi_j^{k^+})^+, (\phi_i^{k^-})^- >_e - \frac{1}{2} < \partial_n(\phi_i^{k^-})^-, (\phi_j^{k^+})^+ >_e - \frac{\gamma}{|e_n|} < (\phi_j^{k^+})^+, (\phi_i^{k^-})^- >_e \end{aligned} \tag{3.15}$$

$$
\begin{aligned}
I(\phi_j^{k^-},\phi_i^{k^+}) = -\sum_e \big(&\frac{1}{2} < \partial_n(\phi_j^{k^-})^+ + \partial_n(\phi_j^{k^-})^-), (\phi_i^{k^+})^+ - (\phi_i^{k^+})^- >_e \\
&+ \frac{1}{2} < \partial_n(\phi_i^{k^+})^+ + \partial_n(\phi_i^{k^+})^-), (\phi_j^{k^-})^+ - (\phi_j^{k^-})^- >_e \\
&- \frac{\gamma}{|e_n|} < (\phi_j^{k^-})^+ - (\phi_j^{k^-})^-, (\phi_i^{k^+})^+ - (\phi_i^{k^+})^- >_e \big) \\
= -&\frac{1}{2} < \partial_n(\phi_j^{k^-})^-, (\phi_i^{k^+})^+ >_e + \frac{1}{2} < \partial_n(\phi_i^{k^+})^+, (\phi_j^{k^-})^- >_e - \frac{\gamma}{|e_n|} < (\phi_j^{k^-})^-, (\phi_i^{k^+})^+ >_e
\end{aligned}
\tag{3.16}
$$

The calculation of each item is relatively complex. We need to do line integration for 2D case and face integration for 3D case. Also, the way we use Gaussian quadratra is different from case to case, depend on the degree of freedom and the dimension. We would rather put this part into the later example explantion.

3.6 BOUNDRY TERM

Bacially, the boundry term is a special case of interior jump term. Since we set all the boundry edge (or face) as the plus edge (or face), so boundry term is actually the plus-plus term. So we have:

Fix $e \in \mathscr{E}_h^D$, there is only one blocks (k^+, k^+) is nonzero for boundary term.Also, all of the formula of the boundry term is very similar to the interior term, include the penalty term and the jump term.

As we now have figure out all the term we need for the stiffness matrix, we then can start to construct it. In later sections several examples will be shown to give a grenral picture of how it works.

4 1D EXAMPLE OF DG-FEM

Overview: In this section , we will show a simple example of DG-FEM to solve a 1D heat equation.

Consider the following 1D-problem, involving a differential equation with boundary conditions:

$$\begin{cases} -u'' = f = -2 \\ u(0) = 0 \\ u(1) = 0 \end{cases} \tag{4.1}$$

Recall the 1D problem (1) in **Section 2.1**. Suppose now the problem is defined on $I = [0,1]$ and we have $f = -2$, $u(0) = 0$ and $u(1) = 1$. The solution can be found explicitly as $u = x(2-x)$, but now we try to solve for the approximation u_h using DG-FEM.

STEP 1. PARTITION THE DOMAIN The accuracy of the approximation as well as the computational cost increase with the number of cells (which are intervals in 1D). In this example, we take number of cells equal to 4 (so $N = 3$): $I_0 = [0, 0.25]$, $I_1 = [0.25, 0.5]$, $I_2 = [0.5, 0.75]$ and $I_3 = [0.75, 1]$.

STEP 2. CHOOSE THE BASIS FUNCTIONS We want each $u_h^{(I_j)}$ to be approximated by a linear combination of a set of basis functions within each interval I_j. These chosen basis functions also take the role as the test function v. We may make our choice of the basis functions to be linear, quadratic or polynomials in higher degree to improve the performance.

Legendre polynomials and lagrange polynomials are some of the popular choices. In this example, the set of linear lagrange polynomials $\hat{\phi}_0 = 1 - x$, $\hat{\phi}_1 = x$ over the **master interval** $[0,1]$ is chosen for use. It would need to be mapped into the corresponding set of functions $\phi_0^{(I_j)}, \phi_1^{(I_j)}$ over the **local interval** I_j for each j afterwards to serve our purpose.

STEP 3. CONSTRUCT THE MATRIX In equation (11) we know that each entry S_{ij} of our stiffness matrix S is defined by the bilinear symmetric function $a(\phi_j, \phi_i)$. In our choice of ϕ, each of the four intervals has 2 degree of freedom. Thus S is a 8×8 matrix, with each parameter of a runs across all $\phi_0^{(I_0)}, \phi_1^{(I_0)}, \phi_0^{(I_1)}, \phi_1^{(I_1)}, \ldots, \phi_0^{(I_3)}, \phi_1^{(I_3)}$.

$$\left[\begin{array}{cc|cc|c} a(\phi_0^{(I_0)},\phi_0^{(I_0)}) & a(\phi_1^{(I_0)},\phi_0^{(I_0)}) & a(\phi_0^{(I_1)},\phi_0^{(I_0)}) & a(\phi_1^{(I_1)},\phi_0^{(I_0)}) & \cdots \\ a(\phi_0^{(I_0)},\phi_1^{(I_0)}) & a(\phi_1^{(I_0)},\phi_1^{(I_0)}) & a(\phi_0^{(I_1)},\phi_1^{(I_0)}) & a(\phi_1^{(I_1)},\phi_1^{(I_0)}) & \\ \hline a(\phi_0^{(I_1)},\phi_0^{(I_1)}) & a(\phi_1^{(I_1)},\phi_0^{(I_1)}) & \ddots & & \\ a(\phi_0^{(I_1)},\phi_1^{(I_1)}) & a(\phi_1^{(I_1)},\phi_1^{(I_1)}) & & & \\ \vdots & & & & \end{array}\right]$$

Note that this **global** matrix is composed by blocks of 2×2 **local** matrices, and each block stores information of basis functions interaction within or across intervals. The four diagonal blocks store those interactions **within an interval**, the six sub-diagonal blocks store those interactions **across adjacent intervals**, and other blocks store zeros – since the basis function values are non-zero only if they are within or on the boundary of their defined intervals.

Recall that a is defined as

$$a(u,v) \equiv \sum_{j=0}^{j=N} \int_{x_j}^{x_{j+1}} u'v' + \sum_{j=0}^{j=N+1} \Big(\{u'\}_j [v]_j + \underbrace{\{v'\}_j [u]_j}_{\text{symmetric term}} \Big) + \underbrace{\gamma \sum_{j=0}^{N+1} \frac{1}{|I_j|} [u]_j [v]_j}_{\text{penalty term}} \tag{4.2}$$

where $\{\cdot\}_j$ is the average of jump on x_j and $[\cdot]_j$ is the jump on x_j as defined in equation (8). Let $k_u, k_v \in \{0,1\}$ be the indices of basis functions within the set that serve the role for the trial function u and the test function v respectively. We now look into the computation of each part of the equation.

STEP 3.1. COMPUTE $\int_{I_j} u'v'$. From the summation sign in equation (9), we know that this term is computed with respect to intervals, so only the diagonal blocks are involved. The term

$$\int_{I_j} \phi_{k_u}^{'(I_j)} \phi_{k_v}^{'(I_j)}$$

may be computed by first considering $\phi_{k_u}^{'(I_j)} \phi_{k_v}^{'(I_j)}$. It can be mapped from $\hat{\phi}'_{k_u} \hat{\phi}'_{k_v}$ with a note to the change in slope of the basis functions from the master interval $[0,1]$ to the local interval I_j.

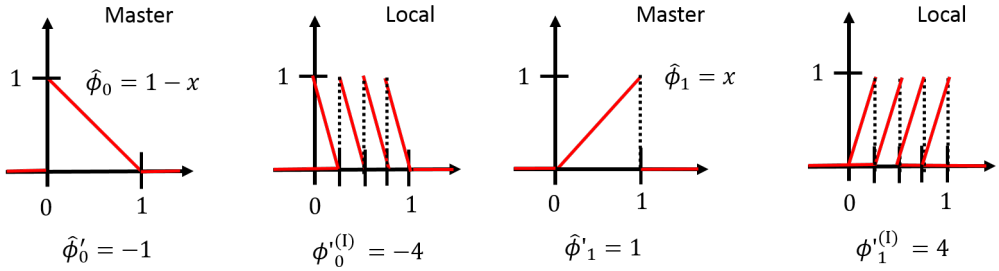


Figure 3.a: Mapping from master interval to local interval

Afterwards integration can be done explicitly or using Gaussian Quadratures. The detail of the calculation is as following:

Let the master cell $\hat{k}$ be the $[0,1]$ segment. Then we have for any segment k, we have

$$\begin{aligned} \int_k \nabla u \nabla v ds &= \int_k \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} ds \\ &= \int_k \frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x}\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} ds \\ &= \frac{|k|}{|\hat{k}|}\int_{\hat{k}} \frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x}\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} d\hat{s} \end{aligned} \tag{4.3}$$

Then we can calculate the value of them. Take the first interval as an example. Then

- $u = \phi_0^{(I_0)}$ $v = \phi_0^{(I_0)}$ $\frac{1}{4}\int_0^1 (-1) * 4 * (-1) * 4 dx = 4$
- $u = \phi_0^{(I_0)}$ $v = \phi_1^{(I_0)}$ $\frac{1}{4}\int_0^1 (-1) * 4 * 1 * 4 dx = -4$
- $u = \phi_1^{(I_0)}$ $v = \phi_0^{(I_0)}$ $\frac{1}{4}\int_0^1 1 * 4 * (-1) * 4 dx = -4$
- $u = \phi_1^{(I_0)}$ $v = \phi_1^{(I_0)}$ $\frac{1}{4}\int_0^1 1 * 4 * 1 * 4 dx = 4$

So the local block is

$$\begin{pmatrix} 4 & -4 \\ -4 & 4 \end{pmatrix} \tag{4.4}$$

STEP 3.2. COMPUTE $\{u'\}_j[v]_j + \{v'\}_j[u]_j$. This term is computed with respect to nodes which stand between adjacent intervals, so for each node x_j both the $(j-1)$-th, j-th diagonal block and the related sub-diagonal blocks are involved. Now since

$$\{\phi'_{k_u}\}_j[\phi_{k_v}]_j + \{\phi'_{k_v}\}_j[\phi_{k_u}]_j$$

is symmetric on ϕ_{k_u}, ϕ_{k_v}, so we may first look at the left term. After expansion we get the expression

$$\begin{aligned}
&\{\phi'_{k_u}\}_j[\phi_{k_v}]_j \\
&= \frac{1}{2}\Big(\phi'_{k_u}(x_j^+) + \phi'_{k_u}(x_j^-)\Big)\Big(\phi_{k_v}(x_j^+) - \phi_{k_v}(x_j^-)\Big) \\
&= \frac{1}{2}\phi'_{k_u}(x_j^+)\phi_{k_v}(x_j^+) - \frac{1}{2}\phi'_{k_u}(x_j^+)\phi_{k_v}(x_j^-) + \frac{1}{2}\phi'_{k_u}(x_j^-)\phi_{k_v}(x_j^+) - \frac{1}{2}\phi'_{k_u}(x_j^-)\phi_{k_v}(x_j^-)
\end{aligned} \tag{4.5}$$

which contains four terms: the "++" term, "+−" term, "−+" and "−−" term respectively.

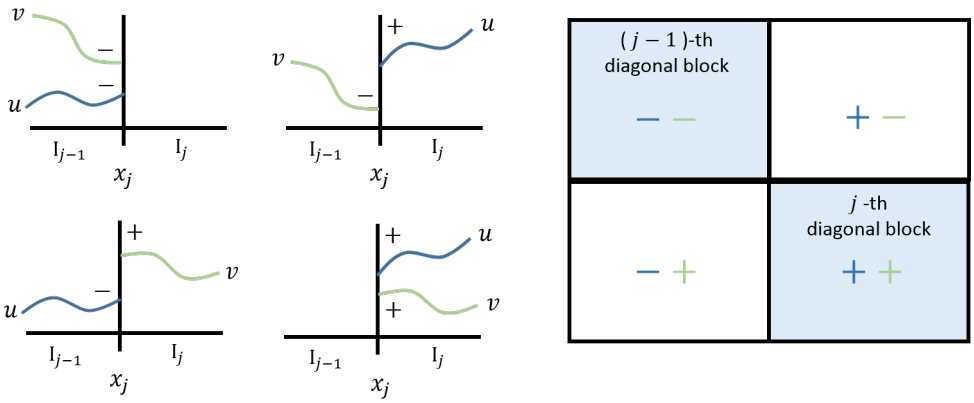


Figure 3.b: the "++" term, "+−" term, "−+" term and "−−" term and their positions

Recall in Figure 2.1.b that x_j^+ and x_j^- denote the right interval and left interval from x_j respectively, thus we may now see that the "++" term is non-zero only if both ϕ'_{k_u} and ϕ_{k_v} are from I_j ; so this term is added to the j-th diagonal block. The "+−" term is non-zero only if ϕ_{k_v} is on I_{j-1} and ϕ'_{k_u} is on I_j; so this term is added to the right sub-diagonal block of the $(j-1)$-th diagonal block. Similarly the "−+" term is added to the left sub-diagonal block of the j-th diagonal block, and the "−−" term is added to the $(j-1)$-th diagonal block.

The expansion of the right term $\{\phi'_{k_u}\}_j[\phi_{k_v}]_j$ is similar and the resultant four terms may be added to the corresponding blocks accordingly with the same principle.

Here, we calculate the u^+v^+ case as an example.

$$\begin{aligned}
&\frac{1}{2}((u^+)'v^+ + (v^+)'u^+) \\
=&\frac{1}{2}(\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x}v^+ + \frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x}u^+) \\
=&2(\frac{\partial u}{\partial \hat{x}}v^+ + \frac{\partial v}{\partial \hat{x}}u^+) \qquad \text{since } \frac{\partial \hat{x}}{\partial x} = 4 \text{ in this case}
\end{aligned} \tag{4.6}$$

- $u^+ = \phi_0^{(I_0)}\ v^+ = \phi_0^{(I_0)}\ 2((-1)*0 + (-1)*0) = 0$
- $u^+ = \phi_0^{(I_0)}\ v^+ = \phi_1^{(I_0)}\ 2(1*0 + (-1)*1) = -2$
- $u^+ = \phi_1^{(I_0)}\ v^+ = \phi_0^{(I_0)}\ 2((-1)*1 + 1*0) = -2$
- $u^+ = \phi_1^{(I_0)}\ v^+ = \phi_1^{(I_0)}\ 2(1*1 + 1*1) = 4$

So the local block is

$$\begin{pmatrix} 0 & -2 \\ -2 & 4 \end{pmatrix} \tag{4.7}$$

STEP 3.3. COMPUTE THE PENALTY TERM. This term is computed with respect to nodes. Again we may expand and get the four "++", "+−", "−+", "−−" terms:

$$\begin{aligned}
&\frac{\gamma}{|I_j|}[\phi_{k_u}]_j[\phi_{k_v}]_j \\
&= \frac{\gamma}{|I_j|}\big(\phi_{k_u}(x_j^+) - \phi_{k_u}(x_j^-)\big)\big(\phi_{k_v}(x_j^+) - \phi_{k_v}(x_j^-)\big) \\
&= \frac{\gamma}{|I_j|}\phi_{k_u}(x_j^+)\phi_{k_v}(x_j^+) - \frac{\gamma}{|I_j|}\phi_{k_u}(x_j^+)\phi_{k_v}(x_j^-) - \frac{\gamma}{|I_j|}\phi_{k_u}(x_j^-)\phi_{k_v}(x_j^+) + \frac{\gamma}{|I_j|}\phi_{k_u}(x_j^-)\phi_{k_v}(x_j^-)
\end{aligned}$$

and added to the corresponding blocks respectively following **Step 3.2**. (note that when ϕ_{k_u}, ϕ_{k_v} come from adjacent intervals, the value $|I_j|$ in the term is computed as the average of $|I_{j-1}|$ and $|I_j|$ instead). The choice of γ is arbitrary – as long as it is "large enough" (see **Section 2.2**) with respect to the degree of polynomial chosen for the basis functions. In this example, we choose $\gamma = 5$.

NOTE 1: AT BOUNDARY NODES. For **Step 3.2** and **3.3**, we have to be more careful when working on the boundary nodes. Clearly at x_0, we have only the "++" term, while at x_{N+1} we have only the "−−" term, since on these two nodes we are working on intervals I_0 and I_N only and their adjacent intervals are not defined .

For the same reason, an "average jump" $\{\cdot\}$ of a function on the boundary node is defined to be the function value at that node itself within the defined interval (while a "jump" $[\cdot]$ is defined as the function value of x_j^+ minus by that of x_j^- for all nodes. Recall that on boundary nodes, we define $x_0^- = x_{N+1}^+ = 0$). For example, at **Step 3.2**, we have the term $\{\phi'_{k_u}\}_0[\phi_{k_v}]_0 = \phi'_{k_u}(x_0^+)\phi_{k_v}(x_0^+)$ and $\{\phi'_{k_u}\}_{N+1}[\phi_{k_v}]_{N+1} = \phi'_{k_u}(x_{N+1}^-)\big(-\phi_{k_v}(x_{N+1}^-)\big)$, which are added to the 0-th diagonal block and N-th diagonal block respectively.

NOTE 2: PRE-COMPUTATION. Before **Step 3**, we may pre-compute the terms $\hat{\phi}_{k_u}\hat{\phi}_{k_v}$, $\hat{\phi}'_{k_u}\hat{\phi}_{k_v}$, $\hat{\phi}'_{k_u}\hat{\phi}'_{k_v}$ for each $k_u, k_v = \{0,1\}$ and for each "++", "+−", "−+", "−−" term on the master interval. They will come into handy when we are working on the bilinear function – since mapping the computed terms from the master interval to local intervals requires only easy linear transformation.

The below shows the actual figures from each step of the matrix construction in our example:

Step 3.1. $\int_{I_j} u'v'$

Local matrices

$$\begin{pmatrix} 4.00 & -4.00 \\ -4.00 & 4.00 \end{pmatrix}$$

$$\begin{bmatrix}
4.00 & -4.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
-4.00 & 4.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 4.00 & -4.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & -4.00 & 4.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 4.00 & -4.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & -4.00 & 4.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 4.00 & -4.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -4.00 & 4.00
\end{bmatrix}$$

Step 3.2. $\{u'\}[v] + \{v'\}[u]$

Local matrices

$$\begin{pmatrix} -8.00 & 4.00 \\ 4.00 & 0.00 \end{pmatrix} \text{ (Boundary node)}$$

$$\begin{bmatrix} 0.00 & 2.00 & -2.00 & 0.00 \\ 2.00 & -4.00 & 4.00 & -2.00 \\ -2.00 & 4.00 & -4.00 & 2.00 \\ 0.00 & -2.00 & 2.00 & 0.00 \end{bmatrix}$$

$$\begin{pmatrix} 0.00 & 4.00 \\ 4.00 & -8.00 \end{pmatrix} \text{ (Boundary node)}$$

$$\begin{bmatrix} -8.00 & 6.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 6.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 6.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & 6.00 & -8.00 \end{bmatrix}$$

Step 3.3. $\frac{\gamma}{|I_j|}\,[u][v]$

Local matrices

$$\begin{pmatrix} 20.00 & 0.00 \\ 0.00 & 0.00 \end{pmatrix} \text{ (Boundary node)}$$

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 20.00 & -20.00 & 0.00 \\ 0.00 & -20.00 & 20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

$$\begin{pmatrix} 0.00 & 0.00 \\ 0.00 & 20.00 \end{pmatrix} \text{ (Boundary node)}$$

$$\begin{bmatrix} 20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 20.00 & -20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -20.00 & 20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 20.00 & -20.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -20.00 & 20.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 20.00 & -20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -20.00 & 20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 20.00 \end{bmatrix}$$

Finish. $a(u, v)$

Final matrix

$$\begin{bmatrix} 16.00 & 2.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 2.00 & 20.00 & -16.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -2.00 & -16.00 & 20.00 & 0.00 & -2.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -2.00 & 0.00 & 20.00 & -16.00 & -2.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -2.00 & -16.00 & 20.00 & 0.00 & -2.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -2.00 & 0.00 & 20.00 & -16.00 & -2.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & -16.00 & 20.00 & 2.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & 2.00 & 16.00 \end{bmatrix}$$

Figure 3.c. Step-by-step illustration of matrix construction

STEP 4. CONSTRUCT THE R.H.S. VECTOR Rewrite equation (11) with the defined notation, we will have the right hand side vector equal to

$$\int_{I_j} f\,\phi_{k_v}^{(I_j)} + \underbrace{\{\phi'_{k_v}\}_j[\phi_{k_u}]_j}_{\text{symmetric term}} + \underbrace{\frac{\gamma}{|I_j|}[\phi_{k_u}]_j[\phi_{k_v}]_j}_{\text{penalty term}} \tag{4.8}$$

which is a 8×1 vector in our example. The integral term is computed with respect to intervals, which can be found explicitly or by Gaussian Quadrature with careful handle of the basis functions mapping from the master intervals to local intervals. For the symmetric term and the penalty term which are to be computed with respect to nodes, notice that we define the jump $[\phi_{k_u}]_j$ to be zero for all internal nodes since u is continuous. Hence we only need to consider those on boundary nodes x_0 and x_{N+1} and add them to the I_0 position and I_N position of the vector accordingly.

Notice that the jump $[\phi_{k_u}]_0$ is essentially $u(a)$ and $[\phi_{k_u}]_{N+1}$ is essentially $-u(b)$. With respect to **Note 1** in **Step 3**, we then obtain our boundary symmetric terms $\phi'_{k_v}(x_0^+)u(a)$ and $\phi'_{k_v}(x_{N+1}^-)\big(-u(b)\big)$,

and also the boundary penalty terms $\frac{\gamma}{|I_0|}u(a)(\phi_{k_v}(x_0^+))$ and $\frac{\gamma}{|I_N|}\big(-u(b)\big)\big(-(\phi_{k_v}(x_{N+1}^-))\big)$. They are added to the I_0-th and I_N-th position of the vector. Substituting everything we would get our final R.H.S. vector

$$[0.25 \quad 0.25 \quad 0.25 \quad 0.25 \quad 0.25 \quad 0.25 \quad 4.25 \quad 16.25]^T$$

STEP 5. SOLVE THE LINEAR SYSTEM. With the given matrix and R.H.S. vector, we may now solve the linear system. In C programming, we may use the dense matrix solver *dgesv* in *LAPACK*, which implements the LU decomposition with partial pivoting and row interchanges:

```
dgesv_(size_Of_Global_Matrix, number_Of_RHS,
pointer_To_Global_Matrix, leading_Dimension_Of_Global_Matrix,
pivot_Indices,
pointer_To_Solution, leading_Dimension_Of_Solution,
info);
```

and our u_h approximation, illustrated by $Matlab$, would be as follow:

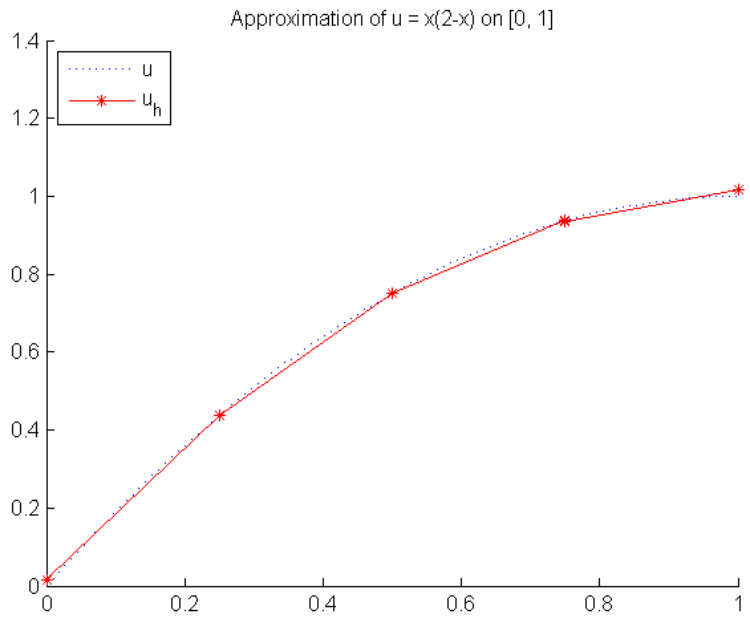


Figure 3.d. Approximation of u using DG-FEM.

5 2D EXAMPLE OF DG-FEM

Overview: In this section , we will show a simple example of DG-FEM to solve a 2D heat equation.

Consider the following 2D-problem, involving a differential equation with boundary conditions:

$$\begin{cases} -\nabla u = f = 0 \\ u = 1 \quad on \quad \partial\Omega \end{cases} \tag{5.1}$$

where the Ω is a square, whose four vertex are $(0,0),(0,1),(1,0)(1,1)$

Just as what we do in the iD case, There are several steps we need to do.

STEP 1. PARTITION THE DOMAIN Here, we divide the mesh into eight equilibrium triangles, where the mesh after division is given as the figure 5.1 shows. For convinence, we number the points, edges and elements. Also, we define the plus and minus side of each esge as well.

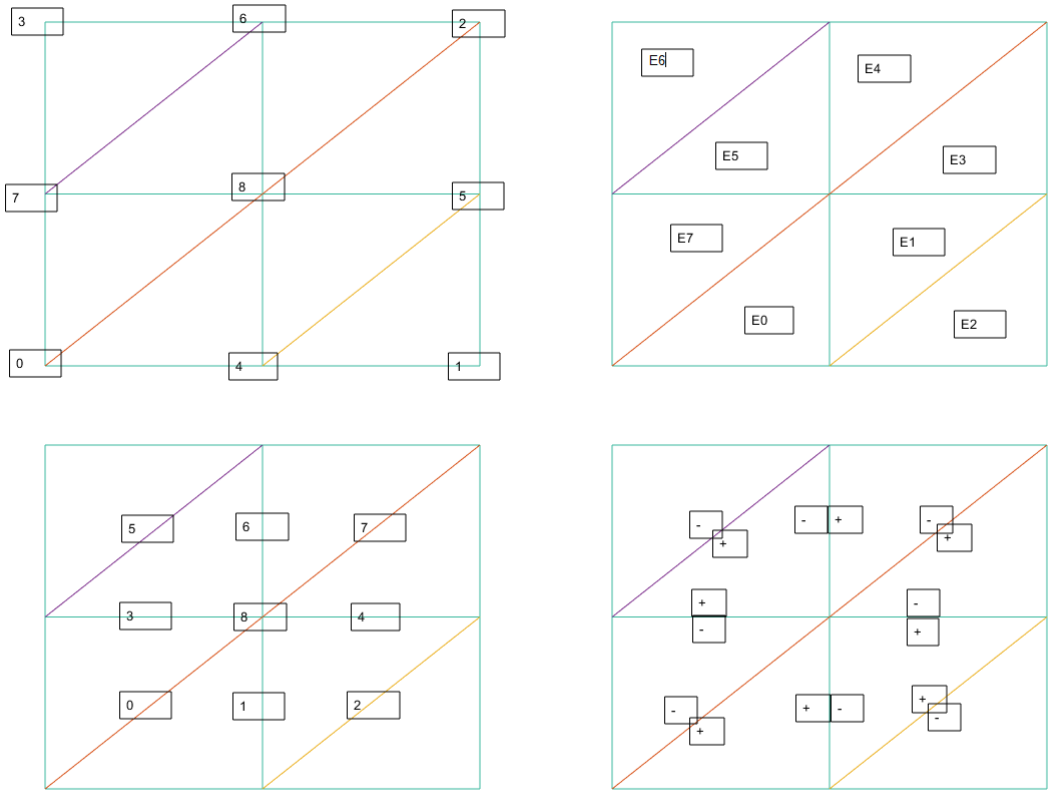


Figure 5.1: The 2D mesh we use

To make it clear to see, we represent the result of partiton as a mesh table.

node and its coordinates		
node index	x	y
0	0	0
1	1	0
2	1	1
3	0	1
4	0.5	0
5	1	0.5
6	0.5	1
7	0	0.5
8	0.5	0.5

cell and its node			
cell index	ndoe1	node2	node3
0	8	4	0
1	4	8	5
2	5	1	4
3	2	5	8
4	8	6	2
5	6	8	7
6	7	3	6
7	0	7	8

Notice here we number the node of each cell clockwisely. Actually you can number the node anyway you like, but in this way the match of quadrature points will be simpler when we start calculation usinng Gaussian quadratra method in integration on edge.

STEP 2. CHOOSE THE BASIS FUNCTIONS We want each $u_h^{(I_j)}$ to be approximated by a linear combination of a set of basis functions within each interval I_j. These chosen basis functions also take the role as the test function v. We may make our choice of the basis functions to be linear, quadratic or polynomials in higher degree to improve the performance.

Legendre polynomials and lagrange polynomials are some of the popular choices. In this example, the set of linear lagrange polynomials $\hat{\phi}_0 = x$, $\hat{\phi}_1 = 1 - x - y$, $\hat{\phi}_2 = y$ over the **master element** is chosen for use. It would need to be mapped into the corresponding set of functions $\phi_0^{(I_j)}, \phi_1^{(I_j)}, \phi_2^{(I_j)}$ over the **local element** I_j for each j afterwards to serve our purpose.

STEP 3. CONSTRUCT THE MATRIX We have known that each entry S_{ij} of our stiffness is defined by the bilinear symmetric function $a(\phi_j, \phi_i)$. So we will calclate the 3 parts, element term, jump term and boundry term one by one. In our choice of ϕ, each of the eight intervals has 3 degree of freedom. Thus S is a 24×24 matrix, with each parameter of a runs across all $\phi_0^{(E_0)}, \phi_1^{(E_0)}, \phi_2^{(E_0)}, \phi_0^{(E_1)}, \phi_1^{(E_1)}, \phi_2^{(E_1)}, \ldots, \phi_0^{(E_3)}, \phi_1^{(E_8)}, \phi_2^{(E_8)}$.

$$
\left[\begin{array}{ccc|ccc|c}
a(\phi_0^{(E_0)},\phi_0^{(E_0)}) & a(\phi_1^{(E_0)},\phi_0^{(E_0)}) & a(\phi_2^{(E_0)},\phi_0^{(E_0)}) & a(\phi_0^{(E_1)},\phi_0^{(E_0)}) & a(\phi_1^{(E_1)},\phi_0^{(E_0)}) & a(\phi_2^{(E_1)},\phi_0^{(E_0)}) & \cdots \\
a(\phi_0^{(E_0)},\phi_1^{(E_0)}) & a(\phi_1^{(E_0)},\phi_1^{(E_0)}) & a(\phi_2^{(E_0)},\phi_1^{(E_0)}) & a(\phi_0^{(E_1)},\phi_1^{(E_0)}) & a(\phi_1^{(E_1)},\phi_1^{(E_0)}) & a(\phi_2^{(E_1)},\phi_1^{(E_0)}) & \\
a(\phi_0^{(E_0)},\phi_2^{(E_0)}) & a(\phi_1^{(E_0)},\phi_2^{(E_0)}) & a(\phi_2^{(E_0)},\phi_2^{(E_0)}) & a(\phi_0^{(E_1)},\phi_2^{(E_0)}) & a(\phi_1^{(E_1)},\phi_2^{(E_0)}) & a(\phi_2^{(E_1)},\phi_2^{(E_0)}) & \\
\hline
a(\phi_0^{(E_0)},\phi_0^{(E_1)}) & a(\phi_1^{(E_0)},\phi_0^{(E_1)}) & a(\phi_2^{(E_0)},\phi_0^{(E_1)}) & \multicolumn{3}{c|}{\multirow{3}{*}{$\ddots$}} & \\
a(\phi_0^{(E_0)},\phi_1^{(E_1)}) & a(\phi_1^{(E_0)},\phi_1^{(E_1)}) & a(\phi_2^{(E_0)},\phi_1^{(E_1)}) & \multicolumn{3}{c|}{} & \\
a(\phi_0^{(E_0)},\phi_2^{(E_1)}) & a(\phi_1^{(E_0)},\phi_2^{(E_1)}) & a(\phi_2^{(E_0)},\phi_2^{(E_1)}) & \multicolumn{3}{c|}{} & \\
\vdots & & & & & & \\
\end{array}\right]
$$

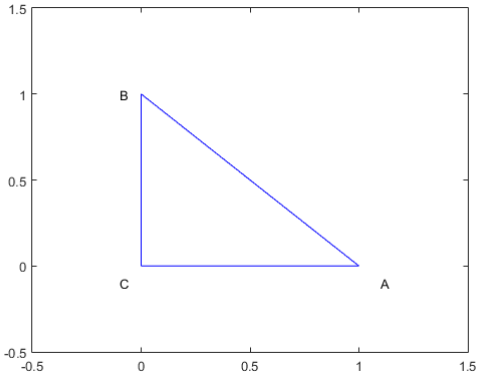


Figure 5.2: Master element

Note that this **global** matrix is composed by blocks of 3×3 **local** matrices, and each block stores information of basis functions interaction within or across elements.

Then we start to compute each part of the equation.

STEP 3.1. COMPUTE $\int_{k_j} \nabla u \nabla v ds$. The $\int_{k_j} \nabla u \nabla v ds$, which we call them 'element term', is a kind of integration over each elements, so only the diagonal blocks are involved. Recall that

$$\begin{aligned}
\int_k \nabla u \nabla v ds &= \int\int (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}) dxdy \\
&= \int\int (\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}) dxdy \\
&= \int\int [(\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y})] |J_{(\hat{x},\hat{y})}| d\hat{x}d\hat{y}
\end{aligned}$$

Hence the term

$$\int_{k_j} \nabla \phi_{k_u}^{(k_j)} \nabla \phi_{k_v}^{(k_j)}$$

can be calculated by mapping. Now we use k_i element as an example. Then the element term of k_i will produce a 3×3 matrix. Note that we have the map from element to k_i:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \frac{1}{|J_{(\hat{x},\hat{y})}|} \begin{pmatrix} y_a - y_c & x_c - x_b \\ y_c - y_a & x_b - x_c \end{pmatrix} * \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \tag{5.2}$$

and

$$\frac{1}{|J_{(\hat{x},\hat{y})}|} \begin{pmatrix} x_a - x_c & x_b - x_c \\ y_a - y_c & y_b - y_c \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{pmatrix} = J \tag{5.3}$$

So we can directly calculate the equation. What we still need to know is how to use Gaussian quadrature for the last step. Here, to get the exact answer of integration of linear function, we need only one quadrature point: $t_1 = (\frac{1}{3}, \frac{1}{3})$, and the weight of it is 1. When the degree of the integrated function becomes higher, we just need to use more quadrature points, according to the gaussian quadrature mathod.

Then we can calculate the value of those functions we used at the quadratra point. For this case, the value we may use is $\frac{\partial u}{\partial \hat{x}}$, $\frac{\partial u}{\partial \hat{y}}$ for $u = \hat{\phi}_0 = x$, $u = \hat{\phi}_1 = 1 - x - y$ and $u = \hat{\phi}_2 = y$. Then we can store the result in the matrix V:

$$V = \begin{pmatrix} \frac{\partial \hat{\phi}_0}{\partial \hat{x}}(t_1) & \frac{\partial \hat{\phi}_0}{\partial \hat{y}}(t_1) \\ \frac{\partial \hat{\phi}_1}{\partial \hat{x}}(t_1) & \frac{\partial \hat{\phi}_1}{\partial \hat{y}}(t_1) \\ \frac{\partial \hat{\phi}_2}{\partial \hat{x}}(t_1) & \frac{\partial \hat{\phi}_2}{\partial \hat{y}}(t_1) \end{pmatrix} \tag{5.4}$$

$Block(jj,kk)$ =0.5(the coefficient of Gaussian quadrature method)∗

$$\frac{\text{area of element}}{\text{area of master cell}} * 1(\text{the weight of quadrature point}) *$$

$((V(kk,1) * J(1,1) + V(kk,2) * J(1,2)) * (V(jj,1) * J(1,1) + V(jj,2) * J(1,2)) +$

$(V(kk,1) * J(2,1) + V(kk,2) * J(2,2)) * (V(jj,1) * J(2,1) + V(jj,2) * J(2,2)));$

Then we can get the local blocks. After putting those blocks into the corresponding diagonal blocks, we can get the global matrix. Notice that since in our mesh, each cell is equlibrium to another, so the value of numbers in element term in each block should be the same (the location of numbers in blocks may be different, depending on the way of mapping).

Take element E0 as an example. The quadrature point we use is $t = (\frac{1}{3}, \frac{1}{3})$. Then we will store all the value of $\frac{\partial \hat{\phi}}{\partial \hat{x}}(t)$ and $\frac{\partial \hat{\phi}}{\partial \hat{y}}(t)$ in V.

$$V = \begin{pmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{pmatrix} \tag{5.5}$$

Also we need to calculate the transform matrix. For E0, the transform matrix J is

$$V = \begin{pmatrix} 0 & 2 \\ -2 & 0 \end{pmatrix} \tag{5.6}$$

Then according to the formula we have given above, we can get the result for the local block:

$$J = \begin{pmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0 \end{pmatrix} \tag{5.7}$$

Columns 1 through 12

0.5000	-0.5000	0	0	0	0	0	0	0	0	0	0
-0.5000	1.0000	-0.5000	0	0	0	0	0	0	0	0	0
0	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
0	0	0	0.5000	-0.5000	0	0	0	0	0	0	0
0	0	0	-0.5000	1.0000	-0.5000	0	0	0	0	0	0
0	0	0	0	-0.5000	0.5000	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0	0	0	0
0	0	0	0	0	0	-0.5000	1.0000	-0.5000	0	0	0
0	0	0	0	0	0	0	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	0	0	0	0.5000	-0.5000	0
0	0	0	0	0	0	0	0	0	-0.5000	1.0000	-0.5000
0	0	0	0	0	0	0	0	0	0	-0.5000	0.5000
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0	0	0	0	0	0	0	0	0	0
-0.5000	1.0000	-0.5000	0	0	0	0	0	0	0	0	0
0	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
0	0	0	0.5000	-0.5000	0	0	0	0	0	0	0
0	0	0	-0.5000	1.0000	-0.5000	0	0	0	0	0	0
0	0	0	0	-0.5000	0.5000	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0	0	0	0
0	0	0	0	0	0	-0.5000	1.0000	-0.5000	0	0	0
0	0	0	0	0	0	0	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	0	0	0	0.5000	-0.5000	0
0	0	0	0	0	0	0	0	0	-0.5000	1.0000	-0.5000
0	0	0	0	0	0	0	0	0	0	-0.5000	0.5000

Figure 5.3: result of element term

STEP 3.2. COMPUTE $\int_{E_i} \partial_n uv + \partial_n vuds$. This term is computed with respect to edges which stand between adjacent element, so for each edge E_i, the two corresponding adjancent element k_m,k_n is included in the calculation. As a result, the m-th, n-th diagonal block and the related sub-diagonal blocks are involved. From section 3.5 we have known how the jump term are divided into 4 parts: u^+v^+, u^-v^-, u^+v^-, u^-v^+. Here we need to discuss them as two cases: the fisrt is the u^+v^+ and u^-v^- case, and the second is the u^+v^- and u^-v^+ case.

First, let's see the u^+v^+ and u^-v^- case. Since here the two functions involved are of the same side, so the result of the calculation will be placed in corresponding diagonal blocks. Now we will focus on how to compute u^+v^+, and u^-v^- are basically the same. Futhermore, notice that in each case, there are two part. The first is the integration of product of the differential of u and another funcion v, which we call it jump term:

$$\pm\frac{1}{2} < \partial_n(\phi_j^{k^-})^-, (\phi_i^{k^+})^+ >_e \pm\frac{1}{2} < \partial_n(\phi_i^{k^+})^+, (\phi_j^{k^-})^- >_e \tag{5.8}$$

The another is the integration of product of these two functions, which we call it **penalty term**.

$$\frac{\gamma}{|e_n|} < (\phi_j^{k^-})^-, (\phi_i^{k^+})^+ >_e \tag{5.9}$$

Although they look similarly, we have to notice they are in different degree: the degree of the functions integrated in the penalty term is higher than the ones in jump term. Since we do the integration by Gaussian quadrature, the degree of the function which is integrated will influence the complexity. Hence we will discuss them separately.

Let's do the jump term first, which has a lower degree and hence simpler as well. The way of mapping is the same as before. Also note that $\int \partial_{\mathbf{n}} uds = \int \nabla u \cdot \mathbf{n}$. So we have

$$\frac{1}{2}\int_{e_i} \nabla u^+ \mathbf{n} v^+ ds = \frac{1}{2}\frac{\text{area of element}}{\text{area of master cell}} \int_{\hat{e}_i} \frac{1}{|J_{(\hat{x},\hat{y})}|} \begin{pmatrix} y_a - y_c & x_c - x_b \\ y_c - y_a & x_b - x_c \end{pmatrix} \nabla\hat{u}\cdot\mathbf{n}\hat{v}d\hat{s} \tag{5.10}$$

In our example, $\nabla\hat{u}$ is always a constant since our basic functions are all linear, and the value of those term in transform matrix is already known. So the equation is transformed into a line integration of $\hat{v}$ on the master edge, which is equal to the value of $\hat{v}$ at the quadrature point on one of the edges of the master cell. Notice which edge should we calculate is depended on the map you choose.

Here we take edge $(0,8)$ as an example. Then the u^+v^+ are only nonzero in the element E0, since the + side of the edge in in element E0. The value of it will be place on the 1-th diagonal blocks. We choose the map from E0 to the master cell, where $[8,4,0]$ of E0 is mapped to $[A,C,B]$ of the master cell. As a result, the edge $(0,8)$ is map to the edge (B,A). So we only need to do the integration of $\hat{v}$ on the edge (B,A). Since in this case, $\hat{v}$ is always linear function, for the basic functions we choose are all linear, we can do the integration with Gaussian quadrature using only one point, that is, the mid point of the edge. The value of it should be calculated in advance and stored in template.

Then we move to the penalty term, which in our example is an integration of a degree 2 function. That means, when use Gaussian quadrature, we need at least 2 quadrature points,$-\sqrt{(3)}/6+0.5$ and $\sqrt{(3)}/6+0.5$ to get the exact answer. Since the function is a product of u and v, we can only store the values of u and v at these quadrature points and then multiply them together. Then there will be a probelm that how should we match these 2 pairs of values. Since we are discussing the u^+v^+ case and the two function are from the same element, they share the same map from the element to master element. As a result, we just match them naturely. That is, the value of u at $-\sqrt{(3)}/6+0.5$ match the value of v at $-\sqrt{(3)}/6+0.5$, and the value of u at $\sqrt{(3)}/6+0.5$ match the value of v at $\sqrt{(3)}/6+0.5$

Now let's look at u^+v^- and u^-v^+ case. Also, we can divide each of these two case to jump term and penalty term. The jump term is just the same with u^+v^+ and u^-v^- case, except for the location of the blocks in the global matrix. When we calculate u^+v^+ and u^-v^- case, we will always put those value in digonal blocks. However, for u^+v^- and u^-v^+, those value will be put in some other blocks. For example, if the edge is the interedge shared by elements k_m and k_n, then the results of u^+v^- and u^-v^+ should be put in (m,n) and (n,m) blocks.

```
Columns 1 through 12

  -0.2500    0.1250   -0.1250         0         0         0         0         0         0         0         0         0
   0.1250   -0.2500    0.3750         0         0         0         0         0         0         0         0         0
  -0.1250    0.3750   -0.2500         0         0         0         0         0         0         0         0         0
        0         0         0   -0.2500    0.3750   -0.1250         0         0         0         0         0         0
        0         0         0    0.3750   -0.2500    0.1250         0         0         0         0         0         0
        0         0         0   -0.1250    0.1250   -0.2500         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0   -0.2500    0.2500   -0.2500
        0         0         0         0         0         0         0         0         0    0.2500         0    0.2500
        0         0         0         0         0         0         0         0         0   -0.2500    0.2500   -0.2500
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0

Columns 13 through 24

        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0   -0.1250    0.1250         0         0         0         0         0         0         0         0         0
  -0.1250   -0.2500    0.1250         0         0         0         0         0         0         0         0         0
   0.1250    0.1250         0         0         0         0         0         0         0         0         0         0
        0         0         0   -0.2500    0.3750   -0.1250         0         0         0         0         0         0
        0         0         0    0.3750   -0.2500    0.1250         0         0         0         0         0         0
        0         0         0   -0.1250    0.1250   -0.2500         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0         0
```

Figure 5.4: result of ++ term

The major difference is the penalty term. Since in u^+v^- and u^-v^+ case, the two functions are from two different elements, so they may use different map. Then there will be a problem of matching the quadrature point. We need to follow steps:

step 1 Get the index of the nodes of the two corresponding elements.

step 2 Figure out the map the two elements use. Usually we set the nodes of each elements are mapped to $[A, B, C]$ of the master node.

step 3 Get the value of the integration on the corresponding edge from the template, and match them according to the map.

For example, suppose the index of the nodes of the two corresponding elements are $[a, b, c]$ and $[b, c, d]$. The interedge is $[b, c]$. In the first element, it is map to the edge $[B, C]$, so we will need the value of the value of the basic functions on edge $[B, C]$, at these two quadrature points. In the second element, it is map to the edge $[A, B]$, so we will need the value of the value of the basic functions on edge $[A, B]$, at these two quadrature points. Then we can match those points. The match rule is: for the first triangle, node 1 is b and node 2 is c; for the second triangle, node 1 is b and node 2 is c (to determine which is the order 1 and which is the order 2, we need to follow the sequence those nodes appears in triangles). Then we match the first value in first element with the first value in second element, and the second value in first element with the second value in second element.

2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.5: result of ++ penalty term

After finish all the calculation of each part of the stiffness matrix, we can combine them and get the global matrix.

Columns 1 through 12

2.0000	2.2500	2.2500	-1.0000	-1.7500	-0.2500	0	0	0	0	0	0
2.2500	1.5000	1.7500	-1.7500	-0.5000	-0.2500	0	0	0	0	0	0
2.2500	1.7500	2.0000	-0.2500	-0.2500	0	0	0	0	0	0	0
-1.0000	-1.7500	-0.2500	2.0000	2.0000	2.0000	-0.5000	-0.5000	-1.5000	0	-0.2500	-0.2500
-1.7500	-0.5000	-0.2500	2.0000	2.0000	2.0000	-0.5000	0	-0.5000	-0.2500	-0.5000	-1.7500
-0.2500	-0.2500	0	2.0000	2.0000	2.0000	-1.5000	-0.5000	-0.5000	-0.2500	-1.7500	-1.0000
0	0	0	-0.5000	-0.5000	-1.5000	2.0000	2.0000	2.5000	0	0	0
0	0	0	-0.5000	0	-0.5000	2.0000	1.0000	2.0000	0	0	0
0	0	0	-1.5000	-0.5000	-0.5000	2.5000	2.0000	2.0000	0	0	0
0	0	0	0	-0.2500	-0.2500	0	0	0	2.0000	1.7500	2.2500
0	0	0	-0.2500	-0.5000	-1.7500	0	0	0	1.7500	1.5000	2.2500
0	0	0	-0.2500	-1.7500	-1.0000	0	0	0	2.2500	2.2500	2.0000
0	0	0	0	0	0	0	0	0	-0.5000	-0.5000	-1.5000
0	0	0	0	0	0	0	0	0	-0.5000	0	-0.5000
0	0	0	0	0	0	0	0	0	-1.5000	-0.5000	-0.5000
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-0.5000	-0.5000	-1.5000	0	0	0	0	0	0	0	0	0
-0.5000	0	-0.5000	0	0	0	0	0	0	0	0	0
-1.5000	-0.5000	-0.5000	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	-0.5000	-0.5000	-1.5000
0	0	0	0	0	0	0	0	0	-0.5000	0	-0.5000
0	0	0	0	0	0	0	0	0	-1.5000	-0.5000	-0.5000
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-0.5000	-0.5000	-1.5000	0	0	0	0	0	0	0	0	0
-0.5000	0	-0.5000	0	0	0	0	0	0	0	0	0
-1.5000	-0.5000	-0.5000	0	0	0	0	0	0	0	0	0
2.0000	2.2500	2.2500	-1.0000	-1.7500	-0.2500	0	0	0	0	0	0
2.2500	1.5000	1.7500	-1.7500	-0.5000	-0.2500	0	0	0	0	0	0
2.2500	1.7500	2.0000	-0.2500	-0.2500	0	0	0	0	0	0	0
-1.0000	-1.7500	-0.2500	2.0000	2.0000	2.0000	-0.5000	-0.5000	-1.5000	0	-0.2500	-0.2500
-1.7500	-0.5000	-0.2500	2.0000	2.0000	2.0000	-0.5000	0	-0.5000	-0.2500	-0.5000	-1.7500
-0.2500	-0.2500	0	2.0000	2.0000	2.0000	-1.5000	-0.5000	-0.5000	-0.2500	-1.7500	-1.0000
0	0	0	-0.5000	-0.5000	-1.5000	2.0000	2.0000	2.5000	0	0	0
0	0	0	-0.5000	0	-0.5000	2.0000	1.0000	2.0000	0	0	0
0	0	0	-1.5000	-0.5000	-0.5000	2.5000	2.0000	2.0000	0	0	0
0	0	0	0	-0.2500	-0.2500	0	0	0	2.0000	1.7500	2.2500
0	0	0	-0.2500	-0.5000	-1.7500	0	0	0	1.7500	1.5000	2.2500
0	0	0	-0.2500	-1.7500	-1.0000	0	0	0	2.2500	2.2500	2.0000

Figure 5.6: the global stiffness matrix

STEP 4. CONSTRUCT THE R.H.S. VECTOR We have the right hand side vector equal to

$$F(v) \equiv (f,v) + \sum_{e_h \in \mathscr{E}_h^D} \left(\frac{\gamma}{|e_h|} < g_D, v >_{e_h} - < g_D, \partial_n v >_{e_h} \right) + \sum_{e_h \in \mathscr{E}_h^N} < g_N, v >_{e_h} \tag{5.11}$$

which is a 24×1 vector in our example. All the part of the R.H.S term can be calculated using the method introduced before. The result of the transpose of the R.H.S vector is

RHS.png

```
Columns 1 through 12

   1     2     3     0     0     0     4     4     4     3     2     1

Columns 13 through 24

   1     2     3     0     0     0     4     4     4     3     2     1
```

Figure 5.7: the transpose of the R.H.S vector

6 3D EXAMPLE OF DG-FEM

Overview: In this section , we will show a simple example of DG-FEM to solve a 3D heat equation. Since all the background knowledge is introduced before, **this chapter will focus on how to calculate everything in programming language.**

Consider the following 2D-problem, involving a differential equation with boundary conditions:

$$\begin{cases} -\nabla u = f = 0 \\ u = 1 \quad on \quad \partial\Omega \end{cases} \tag{6.1}$$

where the Ω is a cube, whose eight vertex are $(0,0,0),(1,0,0),(0,1,0),(1,1,0),(0,0,1),(1,0,1),(0,1,1),(1,1,1)$

Just as what we do in the 2D case, There are several steps we need to do.

STEP 1. PARTITION THE DOMAIN Here, we divide the mesh into six tetrahedrons, where the mesh after division is given as the figure 5.1 shows. For convinence, we number the points, edges and elements. Also, we define the plus and minus side of each esge as well.

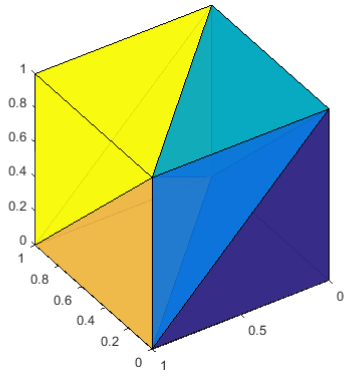


Figure 6.1: The 3D mesh we use

In order to record this mesh into computer, we need to store these kinds of information (which are often given in mesh files, which is produced in mesh generator, like gmsh):

- the coordinates of each node
- the nodes of each element
- the nodes of each face
- the type of each face (boundary face and interior face)
- the side of each interior face (one side is + and the another is -)

In fact, given the first three items we can get the remaining 2 items. The algorithm is like following: First, we have a list of all the elements and a list of all the faces, recording with which nodes they are formed. Notice that each face is a triangle, so they all consist of three node. We start from the the three nodes of the first face in the list, search for whehter there is some element contains those three nodes. Obviously there must be at least one, at most two elements satisfies this condition. The we set the side in the fisrt element is +. If we can successfully find the second element, then the side in it is -, and then the face is a interior face.

To make things easier, here we set our mesh as following:

node and its coordinates			
node index	x	y	z
0	0	0	0
1	1	0	0
2	1	1	0
3	0	1	0
4	1	0	1
5	1	1	1
6	0	1	1
7	0	0	1

cell and its node				
cell index	ndoe1	node2	node3	node4
0	1	2	0	4
1	7	5	4	0
2	5	0	2	4
3	7	0	3	5
4	2	3	0	5
5	6	5	7	3

interior face and its node					
face index	+ element index	-element index	node 1	node 2	node 3
0	0	2	0	4	2
1	1	2	0	5	4
2	1	3	0	7	5
3	4	2	2	5	0
4	5	3	3	5	7
5	4	3	0	5	3

boundary face and its node				
face index	element index	node 1	node 2	node 3
0	0	0	2	1
1	4	0	3	2
2	0	1	2	4
3	2	2	5	4
4	4	2	3	5
5	5	3	6	5
6	0	0	1	4
7	1	0	4	7
8	3	0	7	3
9	5	3	7	6
10	1	5	7	4
11	5	5	6	7

Notice here we number the node of each cell according to the order of map. The first node is mapped to point O in the master tetrahedron. The second node is mapped to point A. The third node is mapped to point B. The last node is mapped to point C. So the map is ensured at this step.

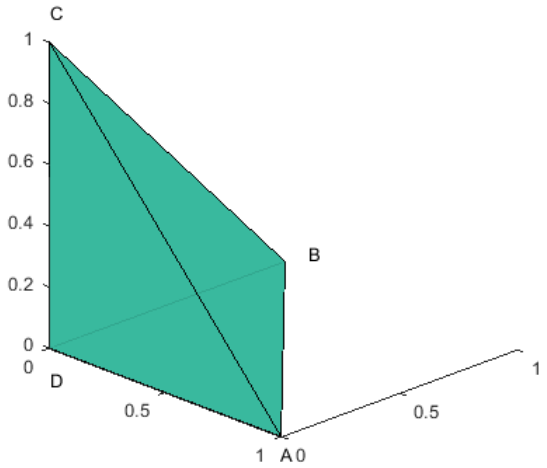


Figure 6.2: The master tetrahedron

STEP 2. CHOOSE THE BASIS FUNCTIONS AND STORE THE TEMPLATE We want each $u_h^{(k_j)}$ to be approximated by a linear combination of a set of basis functions within each element k_j. These chosen basis functions also take the role as the test function v. We may make our choice of the basis functions to be linear, quadratic or polynomials in higher degree to improve the performance.

Legendre polynomials and lagrange polynomials are some of the popular choices. In this example, the set of linear lagrange polynomials $\hat{\phi}_0 = x$, $\hat{\phi}_1 = y$, $\hat{\phi}_2 = z$, $\hat{\phi}_3 = 1 - x - y - z$ over the **master element** is chosen for use. It would need to be mapped into the corresponding set of functions $\phi_0^{(E_j)}, \phi_1^{(E_j)}, \phi_2^{(E_j)}, \phi_2^{(E_j)}$ over the **local element** I_j for each j afterwards to serve our purpose.

After that we need to store our template. The template should include the following values:

- the values of $d\hat{\phi}_0$, $d\hat{\phi}_1$, $d\hat{\phi}_2$. $d\hat{\phi}_3$ at the quadrature points of tetrahedrons in the mester cell
- the values of $d\hat{\phi}_0$, $d\hat{\phi}_1$, $d\hat{\phi}_2$. $d\hat{\phi}_3$ at the quadrature points of triangles in four faces of the mester cell
- the values of $\hat{\phi}_0$, $\hat{\phi}_1$, $\hat{\phi}_2$. $\hat{\phi}_3$ at the quadrature points of triangles in four faces of the mester cell

We will use these value later.

STEP 3. CONSTRUCT THE MATRIX we will calclate the 3 parts, element term, jump term and boundry term one by one. We know that each entry S_{ij} of our stiffness matrix S is defined by the bilinear symmetric function $a(\phi_j, \phi_i)$. In our choice of ϕ, each of the six intervals has 4 degree of freedom. Thus S is a 24×24 matrix, with each parameter of a runs across all $\phi_0^{(E_0)}, \phi_1^{(E_0)}, \phi_2^{(E_0)}, \phi_3^{(E_0)}, \phi_0^{(E_1)}, \ldots, \phi_0^{(E_8)}, \phi_1^{(E_8)}, \phi_2^{(E_8)}, \phi_3^{(E_8)}$.

$$
\left[\begin{array}{cccc|cccc|c}
a(\phi_0^{(E_0)},\phi_0^{(E_0)}) & a(\phi_1^{(E_0)},\phi_0^{(E_0)}) & a(\phi_2^{(E_0)},\phi_0^{(E_0)}) & a(\phi_3^{(E_0)},\phi_0^{(E_0)}) & a(\phi_0^{(E_1)},\phi_0^{(E_0)}) & a(\phi_1^{(E_1)},\phi_0^{(E_0)}) & a(\phi_2^{(E_1)},\phi_0^{(E_0)}) & \cdots \\
a(\phi_0^{(E_0)},\phi_1^{(E_0)}) & a(\phi_1^{(E_0)},\phi_1^{(E_0)}) & a(\phi_2^{(E_0)},\phi_1^{(E_0)}) & a(\phi_3^{(E_0)},\phi_1^{(E_0)}) & a(\phi_0^{(E_1)},\phi_1^{(E_0)}) & a(\phi_1^{(E_1)},\phi_1^{(E_0)}) & a(\phi_2^{(E_1)},\phi_1^{(E_0)}) & \\
a(\phi_0^{(E_0)},\phi_2^{(E_0)}) & a(\phi_1^{(E_0)},\phi_2^{(E_0)}) & a(\phi_2^{(E_0)},\phi_2^{(E_0)}) & a(\phi_3^{(E_0)},\phi_2^{(E_0)}) & a(\phi_0^{(E_1)},\phi_2^{(E_0)}) & a(\phi_1^{(E_1)},\phi_2^{(E_0)}) & a(\phi_2^{(E_1)},\phi_2^{(E_0)}) & \\
a(\phi_0^{(E_0)},\phi_3^{(E_0)}) & a(\phi_1^{(E_0)},\phi_3^{(E_0)}) & a(\phi_2^{(E_0)},\phi_3^{(E_0)}) & a(\phi_3^{(E_0)},\phi_3^{(E_0)}) & a(\phi_0^{(E_1)},\phi_3^{(E_0)}) & a(\phi_1^{(E_1)},\phi_3^{(E_0)}) & a(\phi_2^{(E_1)},\phi_3^{(E_0)}) & \\
\hline
a(\phi_0^{(E_0)},\phi_0^{(E_1)}) & a(\phi_1^{(E_0)},\phi_0^{(E_1)}) & a(\phi_2^{(E_0)},\phi_0^{(E_1)}) & a(\phi_3^{(E_0)},\phi_0^{(E_1)}) & & & & \\
a(\phi_0^{(E_0)},\phi_1^{(E_1)}) & a(\phi_1^{(E_0)},\phi_1^{(E_1)}) & a(\phi_2^{(E_0)},\phi_1^{(E_1)}) & a(\phi_3^{(E_0)},\phi_1^{(E_1)}) & & & & \\
a(\phi_0^{(E_0)},\phi_2^{(E_1)}) & a(\phi_1^{(E_0)},\phi_2^{(E_1)}) & a(\phi_2^{(E_0)},\phi_2^{(E_1)}) & a(\phi_3^{(E_0)},\phi_2^{(E_1)}) & & & & \\
a(\phi_0^{(E_0)},\phi_3^{(E_1)}) & a(\phi_1^{(E_0)},\phi_3^{(E_1)}) & a(\phi_2^{(E_0)},\phi_3^{(E_1)}) & a(\phi_3^{(E_0)},\phi_3^{(E_1)}) & & & & \\
\hline
\vdots & & & & & & & \ddots
\end{array}\right]
$$

Note that this **global** matrix is composed by blocks of 4×4 **local** matrices, and each block stores information of basis functions interaction within or across elements.

Then we start to compute each part of the equation.

STEP 3.1. COMPUTE $\int_{k_j} \nabla u \nabla v$. This term is almost the same as what we have done in 2D case. From Section 3 we know that this term is computed with respect to elementss, so only diagonal blocks are involved. Recall that

$$\begin{aligned}
\int_k \nabla u \nabla v ds &= \int\int (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z})(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial u}{\partial z}) dxdydz \\
&= \int\int (\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} + \frac{\partial u}{\partial z}\frac{\partial v}{\partial z}) dxdydz \\
&= \int\int [(\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial x})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial x}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial y})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial v}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial y}) + \\
&\quad (\frac{\partial u}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial z} + \frac{\partial u}{\partial \hat{y}}\frac{\partial \hat{z}}{\partial z} + \frac{\partial u}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z})(\frac{\partial v}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial z} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z} + \frac{\partial v}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z})]|J_{(\hat{x},\hat{y},\hat{z})}| d\hat{x}d\hat{y}d\hat{z} \\
&= \int\int F(\hat{x},\hat{y},\hat{z}) d\hat{x}d\hat{y}d\hat{z} \\
\text{(Gaussian quadrature)} &= \sum w_i F(\hat{x}_i, \hat{y}_i, \hat{z}_i)
\end{aligned}$$

Hence the term

$$\int_{k_j} \nabla \phi_{k_u}^{(k_j)} \nabla \phi_{k_v}^{(k_j)}$$

can be calculated by mapping. Now we use k_i element as an example. Then the element term of k_i will produce a 4×4 matrix. Note that we have the map from element to k_i:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_a - x_d & x_b - x_d & x_c - x_d \\ y_a - y_d & y_b - y_d & y_c - y_d \\ z_a - z_d & z_b - z_d & z_c - z_d \end{pmatrix} * \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} + \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} \tag{6.2}$$

Then we can get the inverse of

$$\begin{pmatrix} x_a - x_d & x_b - x_d & x_c - x_d \\ y_a - y_d & y_b - y_d & y_c - y_d \\ z_a - z_d & z_b - z_d & z_c - z_d \end{pmatrix} \tag{6.3}$$

which equals to

$$\begin{pmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} & \frac{\partial \hat{x}}{\partial z} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} & \frac{\partial \hat{y}}{\partial z} \\ \frac{\partial \hat{z}}{\partial x} & \frac{\partial \hat{z}}{\partial y} & \frac{\partial \hat{z}}{\partial z} \end{pmatrix} \tag{6.4}$$

So we can directly calculate the equation using the quadrature point $t_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and the weight of it is 1.

The other part is the same with 2D case. All the values we need are stored in template.

Then we can get the local blocks. After putting those blocks into the corresponding diagonal blocks, we can get the global matrix. Notice that since in our mesh, each cell is equlibrium to another, so the value of numbers in element term in each block should be the same (the location of numbers in blocks may be different, depending on the way of mapping).

```
Columns 1 through 12

  0.1667         0         0   -0.1667         0         0         0         0         0         0         0         0
       0    0.1667         0   -0.1667         0         0         0         0         0         0         0         0
       0         0    0.1667   -0.1667         0         0         0         0         0         0         0         0
  -0.1667   -0.1667   -0.1667    0.5000         0         0         0         0         0         0         0         0
       0         0         0         0    0.1667   -0.1667         0         0         0         0         0         0
       0         0         0         0   -0.1667    0.3333         0   -0.1667         0         0         0         0
       0         0         0         0         0         0    0.1667   -0.1667         0         0         0         0
       0         0         0         0         0   -0.1667   -0.1667    0.3333         0         0         0         0
       0         0         0         0         0         0         0         0    0.1667   -0.1667   -0.1667    0.1667
       0         0         0         0         0         0         0         0   -0.1667    0.3333    0.1667   -0.3333
       0         0         0         0         0         0         0         0   -0.1667    0.1667    0.3333   -0.3333
       0         0         0         0         0         0         0         0    0.1667   -0.3333   -0.3333    0.5000
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0

Columns 13 through 24

       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
  0.5000   -0.3333    0.1667   -0.3333         0         0         0         0         0         0         0         0
 -0.3333    0.3333   -0.1667    0.1667         0         0         0         0         0         0         0         0
  0.1667   -0.1667    0.1667   -0.1667         0         0         0         0         0         0         0         0
 -0.3333    0.1667   -0.1667    0.3333         0         0         0         0         0         0         0         0
       0         0         0         0    0.3333   -0.1667         0   -0.1667         0         0         0         0
       0         0         0         0   -0.1667    0.1667         0         0         0         0         0         0
       0         0         0         0         0         0    0.1667   -0.1667         0         0         0         0
       0         0         0         0   -0.1667         0   -0.1667    0.3333         0         0         0         0
       0         0         0         0         0         0         0         0    0.1667         0         0   -0.1667
       0         0         0         0         0         0         0         0         0    0.1667         0   -0.1667
       0         0         0         0         0         0         0         0         0         0    0.1667   -0.1667
       0         0         0         0         0         0         0         0   -0.1667   -0.1667   -0.1667    0.5000
```

Figure 6.3: result of element term

STEP 3.2. COMPUTE $\int_{E_i} \partial_n uv + \partial_n vuds$. This term is computed with respect to faces which stand between adjacent element, so for each Face F_i, the two corresponding adjancent element k_m,k_n is included in the calculation. As a result, the m-th, n-th diagonal block and the related sub-diagonal blocks are involved.

First, let's see the u^+v^+ and u^-v^- case. Since here the two functions involved are of the same side, so the result of the calculation will be placed in corresponding diagonal blocks. Also, we will focus on u^+v^+ only.

Let's do jump term first, which has a lower degree and hence simpler as well. The way of mapping is the same as before. Also note that $\int \partial_{\mathbf{n}} uds = \int \nabla u \cdot \mathbf{n}$. So we have

$$\frac{1}{2}\int_{E_i} \nabla u^+ \mathbf{n} v^+ ds = \frac{1}{2}\frac{\text{volume of element}}{\text{volume of master cell}} \int_{E_i} \nabla \hat{u} \begin{pmatrix} x_a - x_d & x_b - x_d & x_c - x_d \\ y_a - y_d & y_b - y_d & y_c - y_d \\ z_a - z_d & z_b - z_d & z_c - z_d \end{pmatrix}' \cdot \mathbf{n} \hat{v} d\hat{s} \tag{6.5}$$

In our example, $\nabla \hat{u}$ is always a constant, and the value of the jacobian is already known. So the equation is transformed into a integration of $\hat{v}$ on the master edge. Notice which edge should we calculate is depended on the map you choose. The value of it should be calculated in advance and stored in template. So this term is finished.

Then we move to the penalty term, which in our example is an integration of a degree 2 function. That means, when use Gaussian quadrature, we need at least 3 quadrature points,$(1/6,1/6)$, $(1/6,2/3)$ and $(2/3,1/6)$ to get the exact answer. Then since the function is a product of u and v, we can store the values of u and v at these quadrature points and then multiply them together. Since we are discussing the u^+v^+ case and the two function are from the same element, they share the same map from the element to master element. So the value of u at $(1/6,1/6)$ match the value of v at $(1/6,1/6)$, the value of u at $(1/6,2/3)$ match the value of v at $(1/6,2/3)$, and the value of u at $(2/3,1/6)$ match the value of v at $(2/3,1/6)$.

Now let's look at u^+v^- and u^-v^+ case. Also, we can divide each of these two case to jump term and penalty term. The jump term is just the same with u^+v^+ and u^-v^- case, except for the location of the blocks in the global matrix. When we calculate u^+v^+ and u^-v^- case, we will always put those value in digonal blocks. However, for u^+v^- and u^-v^+, those value will be put in some other blocks. For example, if the edge is the interedge shared by elements k_m and k_n, then the results of u^+v^- and u^-v^+ should be put in (m,n) and (n,m) blocks.

The major difference is the penalty term. Since in u^+v^- and u^-v^+ case, the two functions are from two different elements, so they may use different map. Then there will be a problem of matching the quadrature point. In 3D case, these are totally 6 ways of matching. We need to follow the same step as 2D case.

For example, suppose the index of the nodes of the two corresponding elements are $[1,4,7,9]$ and $[7,3,9,4]$. The interedge is $[4,7,9]$. In the first element, it is map to the face $[B,C,D]$, so we will need the value of the value of the basic functions on face $[B,C,D]$, at these three quadrature points. In the second element, it is map to the face $[A,C,D]$, so we will need the value of the value of the basic functions on edge $[A,C,D]$, at these three quadrature points. Then we can match those points. The match rule is: for the first triangle, node 1 is 4, node 2 is 7, node 3 is 9; for the second triangle, node 1 is 7, node 2 is 9, node 3 is 4 (to determine which is the node 1 and which is the node 2, we need to follow the sequence those nodes appears in elements). Then we match the value of node 1 in first element with the value of node 1 in second element, the value of node 2 in first element with the value of node 2 in second element, and the value of node 3 in first element with the value of node 3 in second element.

Columns 1 through 12

0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
-0.2500	-0.2500	-0.2500	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.0833	0.1667	0.0000	0	0	0	0
0	0	0	0	-0.0833	0.1667	0.0000	-0.3333	0	0	0	0
0	0	0	0	0.1667	0.0000	0.1667	-0.0833	0	0	0	0
0	0	0	0	0.0000	-0.3333	-0.0833	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.0833	0.0000	-0.3333	0	0	0	0
0	0	0	0	-0.0833	0.1667	0.1667	0.0000	0	0	0	0
0	0	0	0	0.0000	0.1667	0.1667	-0.0833	0	0	0	0
0	0	0	0	-0.3333	0.0000	-0.0833	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	-0.2500	-0.2500	-0.2500	0

Figure 6.4: result of ++ term

```
penalmat1 =

  Columns 1 through 12

    0.1097    0.0548    0.0548         0         0         0         0         0         0         0         0         0
    0.0548    0.1097    0.0548         0         0         0         0         0         0         0         0         0
    0.0548    0.0548    0.1097         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0    0.1982    0.0496    0.0991    0.0496         0         0         0         0
         0         0         0         0    0.0496    0.0991    0.0496         0         0         0         0         0
         0         0         0         0    0.0991    0.0496    0.1982    0.0496         0         0         0         0
         0         0         0         0    0.0496         0    0.0496    0.0991         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 13 through 24

         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0    0.0991    0.0496    0.0496         0         0         0         0         0
         0         0         0         0    0.0496    0.1982    0.0991    0.0496         0         0         0         0
         0         0         0         0    0.0496    0.0991    0.1982    0.0496         0         0         0         0
         0         0         0         0         0    0.0496    0.0496    0.0991         0         0         0         0
         0         0         0         0         0         0         0         0    0.1097    0.0548    0.0548         0
         0         0         0         0         0         0         0         0    0.0548    0.1097    0.0548         0
         0         0         0         0         0         0         0         0    0.0548    0.0548    0.1097         0
         0         0         0         0         0         0         0         0         0         0         0         0
```

Figure 6.5: result of ++ penalty term

After finish all the calculation of each part of the stiffness matrix, we can combine them and get the global matrix.

Columns 1 through 12

0.4430	0.2215	0.2215	-0.4167	0	0	0	0	-0.1097	0.1952	0.1952	-0.2500
0.2215	0.4430	0.2215	-0.4167	0	0	0	0	-0.0548	0.1952	0.1403	-0.2500
0.2215	0.2215	0.4430	-0.4167	0	0	0	0	-0.0548	0.1403	0.1952	-0.2500
-0.4167	-0.4167	-0.4167	0.5000	0	0	0	0	-0.2500	-0.2500	-0.2500	0
0	0	0	0	0.5315	-0.2004	0.2658	0.0496	0.0338	-0.1667	-0.1824	0.1171
0	0	0	0	-0.2004	0.5991	0.0496	-0.5000	0.0676	-0.1667	-0.0496	0.2004
0	0	0	0	0.2658	0.0496	0.5315	-0.2004	0.1171	-0.1667	-0.0496	0.1509
0	0	0	0	0.0496	-0.5000	-0.2004	0.5991	-0.1667	0	-0.1667	-0.1667
-0.1097	-0.0548	-0.0548	0.2500	-0.1329	-0.2658	-0.2162	0.1667	0.6412	-0.1456	-0.1456	0.5158
-0.3048	-0.3048	-0.3597	0.2500	0.1667	0.1667	0.1667	0	-0.1456	0.7088	0.2215	-0.6171
-0.3048	-0.3597	-0.3048	0.2500	-0.0158	-0.0496	-0.0496	0.1667	-0.1456	0.2215	0.7088	-0.6171
0.2500	0.2500	0.2500	0	-0.2162	-0.2996	-0.3491	0.1667	0.5158	-0.6171	-0.6171	1.3649
0	0	0	0	-0.2996	0.1667	-0.2658	-0.2996	0	0	0	0
0	0	0	0	0.1667	0	0.1667	0.1667	0	0	0	0
0	0	0	0	-0.2658	0.1667	-0.1329	-0.2162	0	0	0	0
0	0	0	0	-0.0496	0.1667	0.0338	-0.0991	0	0	0	0
0	0	0	0	0	0	0	0	-0.1667	-0.1667	0	-0.1667
0	0	0	0	0	0	0	0	0.1171	-0.0496	-0.1667	0.1509
0	0	0	0	0	0	0	0	0.0338	-0.1824	-0.1667	0.1171
0	0	0	0	0	0	0	0	0.0676	-0.0496	-0.1667	0.2004
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.2004	-0.1667	0.0676	-0.0496	0	0	0	0	0	0	0	0
-0.1667	0	-0.1667	-0.1667	0	0	0	0	0	0	0	0
0.0676	-0.1667	0.0338	-0.1329	0	0	0	0	0	0	0	0
0.2004	-0.1667	0.1171	-0.0991	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.2162	-0.1329	-0.2658	0	0	0	0
0	0	0	0	0.1667	-0.0496	-0.0158	-0.0496	0	0	0	0
0	0	0	0	0	0.1667	0.1667	0.1667	0	0	0	0
0	0	0	0	0.1667	-0.3491	-0.2162	-0.2996	0	0	0	0
1.3649	-0.6171	0.5158	-0.6171	-0.3491	-0.2162	-0.2996	0.1667	0.2500	0.2500	0.2500	0
-0.6171	0.7088	-0.1456	0.2215	-0.0496	0.0338	-0.0991	0.1667	-0.3048	-0.3048	-0.3597	0.2500
0.5158	-0.1456	0.6412	-0.1456	-0.2162	-0.1824	-0.2162	0.1667	-0.1097	-0.0548	-0.0548	0.2500
-0.6171	0.2215	-0.1456	0.7088	0.1667	0.1667	0.1667	0	-0.3048	-0.3597	-0.3048	0.2500
0.1509	-0.0496	0.1171	-0.1667	0.5991	-0.2004	0.0496	-0.5000	0	0	0	0
0.1171	-0.1329	-0.0158	-0.1667	-0.2004	0.5315	0.2658	0.0496	0	0	0	0
0.2004	-0.0991	0.1171	-0.1667	0.0496	0.2658	0.5315	-0.2004	0	0	0	0
-0.1667	-0.1667	-0.1667	0	-0.5000	0.0496	-0.2004	0.5991	0	0	0	0
-0.2500	0.1952	-0.0548	0.1403	0	0	0	0	0.5263	0.2632	0.0548	-0.2083
-0.2500	0.1403	-0.0548	0.1952	0	0	0	0	0.2632	0.5263	0.0548	-0.2083
-0.2500	0.1952	-0.1097	0.1952	0	0	0	0	0.0548	0.0548	0.4430	-0.5833
0	-0.2500	-0.2500	-0.2500	0	0	0	0	-0.2083	-0.2083	-0.5833	0.9167

Figure 6.6: the global stiffness matrix

7 FUTURE WORK

After finish calculation of stiffness matrix on 3D DG-FEM, we may continue to work on the followings:

1. PARALLZATION OF 3D DG-FEM At this point all of our procedure are finished in serial code. Implement all the work into parallzation will definitely improve the performance of the programme. We need to consider using parallzation to construct the stiffness matrix with deviding it into several blocks. Also, we will use parallel method to solve the final equation, using trillino.

2. MORE APPLICATIONS. The Poisson's equation is an example that is relatively easy to work on, since it involves only a derivative term. When come to chemical transport phenomena, most often more derivative terms and physical quantities such as time, mass, momentum or energy are involved. Therefore expanding the code to cover different chemical transport equations may be considered.

8 APPENDIX

```
Columns 1 through 12

  0.5000  -0.5000        0        0        0        0        0        0        0        0        0        0
 -0.5000   1.0000  -0.5000        0        0        0        0        0        0        0        0        0
       0  -0.5000   0.5000        0        0        0        0        0        0        0        0        0
       0        0        0   0.5000  -0.5000        0        0        0        0        0        0        0
       0        0        0  -0.5000   1.0000  -0.5000        0        0        0        0        0        0
       0        0        0        0  -0.5000   0.5000        0        0        0        0        0        0
       0        0        0        0        0        0   0.5000  -0.5000        0        0        0        0
       0        0        0        0        0        0  -0.5000   1.0000  -0.5000        0        0        0
       0        0        0        0        0        0        0  -0.5000   0.5000        0        0        0
       0        0        0        0        0        0        0        0        0   0.5000  -0.5000        0
       0        0        0        0        0        0        0        0        0  -0.5000   1.0000  -0.5000
       0        0        0        0        0        0        0        0        0        0  -0.5000   0.5000
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0

Columns 13 through 24

       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
       0        0        0        0        0        0        0        0        0        0        0        0
  0.5000  -0.5000        0        0        0        0        0        0        0        0        0        0
 -0.5000   1.0000  -0.5000        0        0        0        0        0        0        0        0        0
       0  -0.5000   0.5000        0        0        0        0        0        0        0        0        0
       0        0        0   0.5000  -0.5000        0        0        0        0        0        0        0
       0        0        0  -0.5000   1.0000  -0.5000        0        0        0        0        0        0
       0        0        0        0  -0.5000   0.5000        0        0        0        0        0        0
       0        0        0        0        0        0   0.5000  -0.5000        0        0        0        0
       0        0        0        0        0        0  -0.5000   1.0000  -0.5000        0        0        0
       0        0        0        0        0        0        0  -0.5000   0.5000        0        0        0
       0        0        0        0        0        0        0        0        0   0.5000  -0.5000        0
       0        0        0        0        0        0        0        0        0  -0.5000   1.0000  -0.5000
       0        0        0        0        0        0        0        0        0        0  -0.5000   0.5000
```

Figure 8.1: result of element term of 2D case

Columns 1 through 12

-0.2500	0.1250	-0.1250	0	0	0	0	0	0	0	0	0
0.1250	-0.2500	0.3750	0	0	0	0	0	0	0	0	0
-0.1250	0.3750	-0.2500	0	0	0	0	0	0	0	0	0
0	0	0	-0.2500	0.3750	-0.1250	0	0	0	0	0	0
0	0	0	0.3750	-0.2500	0.1250	0	0	0	0	0	0
0	0	0	-0.1250	0.1250	-0.2500	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-0.2500	0.2500	-0.2500
0	0	0	0	0	0	0	0	0	0.2500	0	0.2500
0	0	0	0	0	0	0	0	0	-0.2500	0.2500	-0.2500
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	-0.1250	0.1250	0	0	0	0	0	0	0	0	0
-0.1250	-0.2500	0.1250	0	0	0	0	0	0	0	0	0
0.1250	0.1250	0	0	0	0	0	0	0	0	0	0
0	0	0	-0.2500	0.3750	-0.1250	0	0	0	0	0	0
0	0	0	0.3750	-0.2500	0.1250	0	0	0	0	0	0
0	0	0	-0.1250	0.1250	-0.2500	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.2: result of ++ jump term of 2D case

Columns 1 through 20

2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 21 through 24

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Figure 8.3: result of ++ penalty term of 2D case when r=6

Columns 1 through 12

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.2500	-0.2500	0	0	0	0	0	0
0	0	0	0.2500	0.5000	-0.2500	0	0	0	0	0	0
0	0	0	-0.2500	-0.2500	0	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	-0.5000	0	-0.5000	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.2500	-0.2500
0	0	0	0	0	0	0	0	0	-0.2500	0.5000	0.2500
0	0	0	0	0	0	0	0	0	-0.2500	0.2500	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
-0.5000	0	-0.5000	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
0	0	0	0	0.2500	-0.2500	0	0	0	0	0	0
0	0	0	0.2500	0.5000	-0.2500	0	0	0	0	0	0
0	0	0	-0.2500	-0.2500	0	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	-0.5000	0	-0.5000	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	0	0
0	0	0	0	0	0	0	0	0	0.5000	-0.7500	0.2500
0	0	0	0	0	0	0	0	0	-0.7500	0.5000	-0.2500
0	0	0	0	0	0	0	0	0	0.2500	-0.2500	0.5000

Figure 8.4: result of – jump term of 2D case

Columns 1 through 20

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 21 through 24

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	0	0	0
0	0	0	0
1	0	0	0
0	1	0	2
0	0	1	2
0	2	2	2

Figure 8.5: result of – penalty term of 2D case when r=6

```
 Columns 1 through 12

         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0    0.2500   -0.2500         0         0         0         0         0         0         0         0         0
    0.2500    0.5000   -0.2500         0         0         0         0         0         0         0         0         0
   -0.2500   -0.2500         0         0         0         0         0         0         0         0         0         0
         0         0         0    0.5000   -0.5000    0.5000         0         0         0         0         0         0
         0         0         0   -0.5000         0   -0.5000         0         0         0         0         0         0
         0         0         0    0.5000   -0.5000    0.5000         0         0         0         0         0         0
         0         0         0         0   -0.2500   -0.2500         0         0         0         0         0         0
         0         0         0   -0.2500    0.5000    0.2500         0         0         0         0         0         0
         0         0         0   -0.2500    0.2500         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0    0.5000   -0.5000    0.5000
         0         0         0         0         0         0         0         0         0   -0.5000         0   -0.5000
         0         0         0         0         0         0         0         0         0    0.5000   -0.5000    0.5000
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
    0.5000   -0.5000    0.5000         0         0         0         0         0         0         0         0         0
   -0.5000         0   -0.5000         0         0         0         0         0         0         0         0         0
    0.5000   -0.5000    0.5000         0         0         0         0         0         0         0         0         0

 Columns 13 through 24

         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0    0.2500   -0.2500         0         0         0         0         0         0         0         0         0
    0.2500    0.5000   -0.2500         0         0         0         0         0         0         0         0         0
   -0.2500   -0.2500         0         0         0         0         0         0         0         0         0         0
         0         0         0    0.5000   -0.5000    0.5000         0         0         0         0         0         0
         0         0         0   -0.5000         0   -0.5000         0         0         0         0         0         0
         0         0         0    0.5000   -0.5000    0.5000         0         0         0         0         0         0
         0         0         0         0   -0.2500   -0.2500         0         0         0         0         0         0
         0         0         0   -0.2500    0.5000    0.2500         0         0         0         0         0         0
         0         0         0   -0.2500    0.2500         0         0         0         0         0         0         0
```

Figure 8.6: result of +- jump term of 2D case

Columns 1 through 20

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0
1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0

Columns 21 through 24

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Figure 8.7: result of – penalty term of 2D case when r=6

Columns 1 through 12

0	0	0	0	0.2500	-0.2500	0	0	0	0	0	0
0	0	0	0.2500	0.5000	-0.2500	0	0	0	0	0	0
0	0	0	-0.2500	-0.2500	0	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	-0.2500	-0.2500
0	0	0	0	0	0	-0.5000	0	-0.5000	-0.2500	0.5000	0.2500
0	0	0	0	0	0	0.5000	-0.5000	0.5000	-0.2500	0.2500	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0.5000	-0.5000	0.5000
0	0	0	0	0	0	0	0	0	-0.5000	0	-0.5000
0	0	0	0	0	0	0	0	0	0.5000	-0.5000	0.5000
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
-0.5000	0	-0.5000	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0.5000	0	0	0	0	0	0	0	0	0
0	0	0	0	0.2500	-0.2500	0	0	0	0	0	0
0	0	0	0.2500	0.5000	-0.2500	0	0	0	0	0	0
0	0	0	-0.2500	-0.2500	0	0	0	0	0	0	0
0	0	0	0	0	0	0.5000	-0.5000	0.5000	0	-0.2500	-0.2500
0	0	0	0	0	0	-0.5000	0	-0.5000	-0.2500	0.5000	0.2500
0	0	0	0	0	0	0.5000	-0.5000	0.5000	-0.2500	0.2500	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.8: result of +- jump term of 2D case

Columns 1 through 20

```
     0     0     0     1     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     1     0     2     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     1     2     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     2     0     1     0     2     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1     0     2     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     2     0     1     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     2     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     2     1     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     2     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Columns 21 through 24

```
     0     1     0     2
     0     0     0     0
     0     2     0     1
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     2     0     0     0
     0     0     1     2
     1     0     2     1
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

Figure 8.9: result of -+ penalty term of 2D case when r=6

Columns 1 through 12

0	0.5000	0.5000	0	0	0	0	0	0	0	0	0
0.5000	-1.0000	-0.5000	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.0000	0	0	0
0	0	0	0	0	0	0	-2.0000	0	0	0	0
0	0	0	0	0	0	1.0000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.5000	0.5000
0	0	0	0	0	0	0	0	0	-0.5000	-1.0000	0.5000
0	0	0	0	0	0	0	0	0	0.5000	0.5000	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0.5000	0.5000	0	0	0	0	0	0	0	0	0
0.5000	-1.0000	-0.5000	0	0	0	0	0	0	0	0	0
0.5000	-0.5000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.0000	0	0	0
0	0	0	0	0	0	0	-2.0000	0	0	0	0
0	0	0	0	0	0	1.0000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.5000	0.5000
0	0	0	0	0	0	0	0	0	-0.5000	-1.0000	0.5000
0	0	0	0	0	0	0	0	0	0.5000	0.5000	0

Figure 8.10: result of boundary jump term of 2D case

Columns 1 through 20

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 21 through 24

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
2	0	0	0
1	0	0	0
0	1	2	0
0	2	1	0
0	0	0	0

Figure 8.11: result of boundary penalty term of 2D case when r=6

```
Columns 1 through 12

  0.1667         0         0   -0.1667         0         0         0         0         0         0         0         0
       0    0.1667         0   -0.1667         0         0         0         0         0         0         0         0
       0         0    0.1667   -0.1667         0         0         0         0         0         0         0         0
  -0.1667   -0.1667   -0.1667    0.5000         0         0         0         0         0         0         0         0
       0         0         0         0    0.1667   -0.1667         0         0         0         0         0         0
       0         0         0         0   -0.1667    0.3333         0   -0.1667         0         0         0         0
       0         0         0         0         0         0    0.1667   -0.1667         0         0         0         0
       0         0         0         0         0   -0.1667   -0.1667    0.3333         0         0         0         0
       0         0         0         0         0         0         0         0    0.1667   -0.1667   -0.1667    0.1667
       0         0         0         0         0         0         0         0   -0.1667    0.3333    0.1667   -0.3333
       0         0         0         0         0         0         0         0   -0.1667    0.1667    0.3333   -0.3333
       0         0         0         0         0         0         0         0    0.1667   -0.3333   -0.3333    0.5000
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0

Columns 13 through 24

       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0         0         0
  0.5000   -0.3333    0.1667   -0.3333         0         0         0         0         0         0         0         0
 -0.3333    0.3333   -0.1667    0.1667         0         0         0         0         0         0         0         0
  0.1667   -0.1667    0.1667   -0.1667         0         0         0         0         0         0         0         0
 -0.3333    0.1667   -0.1667    0.3333         0         0         0         0         0         0         0         0
       0         0         0         0    0.3333   -0.1667         0   -0.1667         0         0         0         0
       0         0         0         0   -0.1667    0.1667         0         0         0         0         0         0
       0         0         0         0         0         0    0.1667   -0.1667         0         0         0         0
       0         0         0         0   -0.1667         0   -0.1667    0.3333         0         0         0         0
       0         0         0         0         0         0         0         0    0.1667         0         0   -0.1667
       0         0         0         0         0         0         0         0         0    0.1667         0   -0.1667
       0         0         0         0         0         0         0         0         0         0    0.1667   -0.1667
       0         0         0         0         0         0         0         0   -0.1667   -0.1667   -0.1667    0.5000
```

Figure 8.12: result of eleterm term of 3D case

Columns 1 through 12

0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
0.1667	0.1667	0.1667	-0.2500	0	0	0	0	0	0	0	0
-0.2500	-0.2500	-0.2500	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.0833	0.1667	0.0000	0	0	0	0
0	0	0	0	-0.0833	0.1667	0.0000	-0.3333	0	0	0	0
0	0	0	0	0.1667	0.0000	0.1667	-0.0833	0	0	0	0
0	0	0	0	0.0000	-0.3333	-0.0833	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.0833	0.0000	-0.3333	0	0	0	0
0	0	0	0	-0.0833	0.1667	0.1667	0.0000	0	0	0	0
0	0	0	0	0.0000	0.1667	0.1667	-0.0833	0	0	0	0
0	0	0	0	-0.3333	0.0000	-0.0833	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	-0.2500
0	0	0	0	0	0	0	0	-0.2500	-0.2500	-0.2500	0

Figure 8.13: result of ++ jump term of 3D case

```
penalmat1 =

  Columns 1 through 12

    0.1097    0.0548    0.0548         0         0         0         0         0         0         0         0         0
    0.0548    0.1097    0.0548         0         0         0         0         0         0         0         0         0
    0.0548    0.0548    0.1097         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0    0.1982    0.0496    0.0991    0.0496         0         0         0         0
         0         0         0         0    0.0496    0.0991    0.0496         0         0         0         0         0
         0         0         0         0    0.0991    0.0496    0.1982    0.0496         0         0         0         0
         0         0         0         0    0.0496         0    0.0496    0.0991         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 13 through 24

         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0    0.0991    0.0496    0.0496         0         0         0         0         0
         0         0         0         0    0.0496    0.1982    0.0991    0.0496         0         0         0         0
         0         0         0         0    0.0496    0.0991    0.1982    0.0496         0         0         0         0
         0         0         0         0         0    0.0496    0.0496    0.0991         0         0         0         0
         0         0         0         0         0         0         0         0    0.1097    0.0548    0.0548         0
         0         0         0         0         0         0         0         0    0.0548    0.1097    0.0548         0
         0         0         0         0         0         0         0         0    0.0548    0.0548    0.1097         0
         0         0         0         0         0         0         0         0         0         0         0         0
```

Figure 8.14: result of ++ penalty term of 3D case when r=1

```
Columns 1 through 12
```

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-0.1667	0.0833	0.0833	-0.2500
0	0	0	0	0	0	0	0	0.0833	-0.1667	-0.0000	0.3333
0	0	0	0	0	0	0	0	0.0833	-0.0000	-0.1667	0.3333
0	0	0	0	0	0	0	0	-0.2500	0.3333	0.3333	-0.6667
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

```
Columns 13 through 24
```

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-0.6667	0.3333	-0.2500	0.3333	0	0	0	0	0	0	0	0
0.3333	-0.1667	0.0833	0	0	0	0	0	0	0	0	0
-0.2500	0.0833	-0.1667	0.0833	0	0	0	0	0	0	0	0
0.3333	0	0.0833	-0.1667	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.15: result of – jump term of 3D case

Columns 1 through 12

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.3079	0.1044	0.1044	0.0991
0	0	0	0	0	0	0	0	0.1044	0.2088	0.0548	0.0496
0	0	0	0	0	0	0	0	0.1044	0.0548	0.2088	0.0496
0	0	0	0	0	0	0	0	0.0991	0.0496	0.0496	0.1982
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.1982	0.0496	0.0991	0.0496	0	0	0	0	0	0	0	0
0.0496	0.2088	0.1044	0.0548	0	0	0	0	0	0	0	0
0.0991	0.1044	0.3079	0.1044	0	0	0	0	0	0	0	0
0.0496	0.0548	0.1044	0.2088	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.16: result of – penalty term of 3D case when r=1

Columns 1 through 12

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.0000	0.2500	-0.0833	-0.1667	-0.1667	0.1667	0	0	0	0
-0.2500	-0.2500	-0.2500	0.2500	0.1667	0.1667	0.1667	0	0	0	0	0
-0.2500	-0.2500	-0.2500	0.2500	0.0833	0	0.0000	0.1667	0	0	0	0
0.2500	0.2500	0.2500	0	-0.1667	-0.2500	-0.2500	0.1667	0	0	0	0
0	0	0	0	-0.2500	0.1667	-0.1667	-0.2500	0	0	0	0
0	0	0	0	0.1667	0	0.1667	0.1667	0	0	0	0
0	0	0	0	-0.1667	0.1667	-0.0833	-0.1667	0	0	0	0
0	0	0	0	-0.0000	0.1667	0.0833	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	-0.1667	-0.0833	-0.1667	0	0	0	0
0	0	0	0	0.1667	0	0.0833	0.0000	0	0	0	0
0	0	0	0	0	0.1667	0.1667	0.1667	0	0	0	0
0	0	0	0	0.1667	-0.2500	-0.1667	-0.2500	0	0	0	0
0	0	0	0	-0.2500	-0.1667	-0.2500	0.1667	0.2500	0.2500	0.2500	0
0	0	0	0	0	0.0833	0.0000	0.1667	-0.2500	-0.2500	-0.2500	0.2500
0	0	0	0	-0.1667	-0.0833	-0.1667	0.1667	0	0	0.0000	0.2500
0	0	0	0	0.1667	0.1667	0.1667	0	-0.2500	-0.2500	-0.2500	0.2500
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.17: result of +- jump term of 3D case

Columns 1 through 12

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.1097	0.0548	0.0548	0	0.0496	0.0991	0.0496	0	0	0	0	0
0.0548	0.0548	0.1097	0	0	0	0	0	0	0	0	0
0.0548	0.1097	0.0548	0	0.0991	0.0496	0.0496	0	0	0	0	0
0	0	0	0	0.0496	0.0496	0.0991	0	0	0	0	0
0	0	0	0	0.0496	0	0.0991	0.0496	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.0991	0	0.0496	0.0496	0	0	0	0
0	0	0	0	0.0496	0	0.0496	0.0991	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.0496	0.0496	0.0991	0	0	0	0
0	0	0	0	0	0.0496	0.0991	0.0496	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.0991	0.0496	0.0496	0	0	0	0
0	0	0	0	0.0991	0.0496	0.0496	0	0	0	0	0
0	0	0	0	0.0496	0.0496	0.0991	0	0.0548	0.0548	0.1097	0
0	0	0	0	0.0496	0.0991	0.0496	0	0.1097	0.0548	0.0548	0
0	0	0	0	0	0	0	0	0.0548	0.1097	0.0548	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.18: result of +- penalty term of 3D case when r=1

Columns 1 through 12

0	0	0	0	0	0	0	0	0	0.2500	0.2500	-0.2500
0	0	0	0	0	0	0	0	0	0.2500	0.2500	-0.2500
0	0	0	0	0	0	0	0	-0.0000	0.2500	0.2500	-0.2500
0	0	0	0	0	0	0	0	-0.2500	-0.2500	-0.2500	0
0	0	0	0	0	0	0	0	0.0833	-0.1667	-0.0833	0.1667
0	0	0	0	0	0	0	0	0.1667	-0.1667	0	0.2500
0	0	0	0	0	0	0	0	0.1667	-0.1667	-0.0000	0.2500
0	0	0	0	0	0	0	0	-0.1667	0	-0.1667	-0.1667
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-0.1667	-0.1667	0	-0.1667
0	0	0	0	0	0	0	0	0.1667	0	-0.1667	0.2500
0	0	0	0	0	0	0	0	0.0833	-0.0833	-0.1667	0.1667
0	0	0	0	0	0	0	0	0.1667	-0.0000	-0.1667	0.2500
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.2500	-0.1667	0.1667	0.0000	0	0	0	0	0	0	0	0
-0.1667	0	-0.1667	-0.1667	0	0	0	0	0	0	0	0
0.1667	-0.1667	0.0833	-0.0833	0	0	0	0	0	0	0	0
0.2500	-0.1667	0.1667	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.2500	0	0.1667	-0.1667	0	0	0	0	0	0	0	0
0.1667	-0.0833	0.0833	-0.1667	0	0	0	0	0	0	0	0
0.2500	-0.0000	0.1667	-0.1667	0	0	0	0	0	0	0	0
-0.1667	-0.1667	-0.1667	0	0	0	0	0	0	0	0	0
-0.2500	0.2500	0	0.2500	0	0	0	0	0	0	0	0
-0.2500	0.2500	0	0.2500	0	0	0	0	0	0	0	0
-0.2500	0.2500	-0.0000	0.2500	0	0	0	0	0	0	0	0
0	-0.2500	-0.2500	-0.2500	0	0	0	0	0	0	0	0

Figure 8.19: result of -+ jump term of 3D case

Columns 1 through 12

0	0	0	0	0	0	0	0	0.1097	0.0548	0.0548	0
0	0	0	0	0	0	0	0	0.0548	0.0548	0.1097	0
0	0	0	0	0	0	0	0	0.0548	0.1097	0.0548	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.0496	0	0.0991	0.0496
0	0	0	0	0	0	0	0	0.0991	0	0.0496	0.0496
0	0	0	0	0	0	0	0	0.0496	0	0.0496	0.0991
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.0496	0.0496	0	0.0991
0	0	0	0	0	0	0	0	0.0496	0.0991	0	0.0496
0	0	0	0	0	0	0	0	0.0991	0.0496	0	0.0496
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.0496	0	0.0991	0.0496	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.0991	0	0.0496	0.0496	0	0	0	0	0	0	0	0
0.0496	0	0.0496	0.0991	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.0991	0.0496	0.0496	0	0	0	0	0	0	0	0	0
0.0496	0.0496	0.0991	0	0	0	0	0	0	0	0	0
0.0496	0.0991	0.0496	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0.0548	0.0548	0.1097	0	0	0	0	0	0	0	0
0	0.1097	0.0548	0.0548	0	0	0	0	0	0	0	0
0	0.0548	0.1097	0.0548	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.20: result of -+ penalty term of 3D case when r=1

Columns 1 through 12

0	-0.3333	-0.3333	0.1667	0	0	0	0	0	0	0	0
-0.3333	0	-0.3333	0.1667	0	0	0	0	0	0	0	0
-0.3333	-0.3333	0	0.1667	0	0	0	0	0	0	0	0
0.1667	0.1667	0.1667	1.0000	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.1667	-0.3333	-0.0000	0	0	0	0
0	0	0	0	-0.1667	0.3333	0	0.3333	0	0	0	0
0	0	0	0	-0.3333	0	0	-0.1667	0	0	0	0
0	0	0	0	-0.0000	0.3333	-0.1667	0.3333	0	0	0	0
0	0	0	0	0	0	0	0	0	-0.1667	-0.1667	-0.1667
0	0	0	0	0	0	0	0	-0.1667	0.3333	0.3333	0.0000
0	0	0	0	0	0	0	0	-0.1667	0.3333	0.3333	0.0000
0	0	0	0	0	0	0	0	-0.1667	0.0000	0.0000	-0.3333
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-0.3333	0	-0.1667	-0.0000	0	0	0	0	0	0	0	0
0	0.3333	-0.1667	0.3333	0	0	0	0	0	0	0	0
-0.1667	-0.1667	0	-0.1667	0	0	0	0	0	0	0	0
-0.0000	0.3333	-0.1667	0.3333	0	0	0	0	0	0	0	0
0	0	0	0	0.3333	-0.1667	0	0.3333	0	0	0	0
0	0	0	0	-0.1667	0	-0.3333	-0.0000	0	0	0	0
0	0	0	0	0	-0.3333	0	-0.1667	0	0	0	0
0	0	0	0	0.3333	-0.0000	-0.1667	0.3333	0	0	0	0
0	0	0	0	0	0	0	0	0	-0.3333	-0.3333	0.1667
0	0	0	0	0	0	0	0	-0.3333	0	-0.3333	0.1667
0	0	0	0	0	0	0	0	-0.3333	-0.3333	0	0.1667
0	0	0	0	0	0	0	0	0.1667	0.1667	0.1667	1.0000

Figure 8.21: result of boundary jump term of 3D case

Columns 1 through 12

0.1667	0.0417	0.0417	0.0833	0	0	0	0	0	0	0	0
0.0417	0.1667	0.0417	0.0833	0	0	0	0	0	0	0	0
0.0417	0.0417	0.1667	0.0833	0	0	0	0	0	0	0	0
0.0833	0.0833	0.0833	0.2500	0	0	0	0	0	0	0	0
0	0	0	0	0.0833	0.0417	0	0.0417	0	0	0	0
0	0	0	0	0.0417	0.1667	0.0417	0.0833	0	0	0	0
0	0	0	0	0	0.0417	0.0833	0.0417	0	0	0	0
0	0	0	0	0.0417	0.0833	0.0417	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.0833	0.0417	0.0417
0	0	0	0	0	0	0	0	0	0.0417	0.0833	0.0417
0	0	0	0	0	0	0	0	0	0.0417	0.0417	0.0833
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

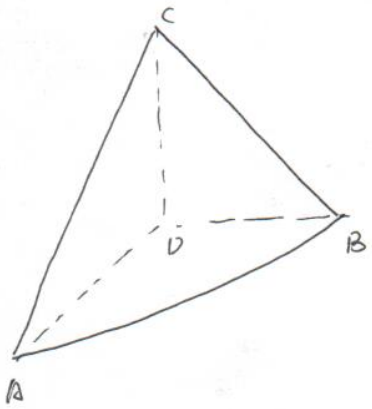
Columns 13 through 24

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.0833	0.0417	0	0.0417	0	0	0	0	0	0	0	0
0.0417	0.0833	0	0.0417	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.0417	0.0417	0	0.0833	0	0	0	0	0	0	0	0
0	0	0	0	0.1667	0.0417	0.0417	0.0833	0	0	0	0
0	0	0	0	0.0417	0.0833	0	0.0417	0	0	0	0
0	0	0	0	0.0417	0	0.0833	0.0417	0	0	0	0
0	0	0	0	0.0833	0.0417	0.0417	0.1667	0	0	0	0
0	0	0	0	0	0	0	0	0.1667	0.0417	0.0417	0.0833
0	0	0	0	0	0	0	0	0.0417	0.1667	0.0417	0.0833
0	0	0	0	0	0	0	0	0.0417	0.0417	0.1667	0.0833
0	0	0	0	0	0	0	0	0.0833	0.0833	0.0833	0.2500

Figure 8.22: result of boundary penalty term of 3D case when r=1

Detail about the mapping problem. 3D

① template about the gaussian quadrature



There are four faces in master cell

face 0	B C D
face 1	A C D
face 2	A B D
face 3	A B C

There are 3 quadrature pts for face integral of degree 2 function:

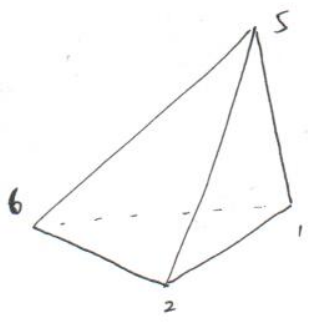
Pt 0	$(\frac{1}{6}, \frac{1}{6})$
Pt 1	$(\frac{1}{6}, \frac{2}{3})$
Pt 2	$(\frac{2}{3}, \frac{1}{6})$

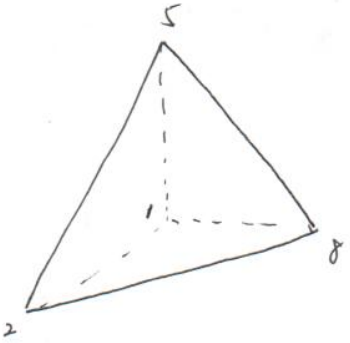
Since we are solving integration like $\int_T f(x)\,g(x)$ where f, g are constant or linear functions.

Then our template stores the value of f, g at these 3 points on the 4 faces.

② matching problem: 3D

There is two elements.





Z_1: [6, 1, 2, 5] Z_2: [5, 1, 2, 8]

They are all mapped to [A, B, C, D] defaultly.

So the common face [1, 2, 5] maps to [B, C, D] for Z_1

maps to [A, B, C] for Z_2

So for $\int_F f^{(E1)}(x)\, g^{(E2)}(x)$, we get the template value of f on [B, C, D] $\Leftrightarrow F_0$

template value of g on [A, B, C] $\Leftrightarrow \bar{F}_3$

Then we have $V_{f_0}^{(0)}$, $V_{f_0}^{(1)}$, $V_{f_0}^{(2)}$,

$V_{g_3}^{(0)}$, $V_{g_3}^{(1)}$, $V_{g_3}^{(2)}$ as template values.

	node 0	node 1	node 2
Z_1	1	2	5
Z_2	5	1	2

by the order showed in the element

We can see the match case is 3D

0 1 2 → 1 2 0

In general, there is 6 matching cases

index			
0	0	1	2
1	0	2	1
2	1	0	2
3	1	2	0
4	2	0	1
5	2	1	0

So our case is the index 3. —— 120

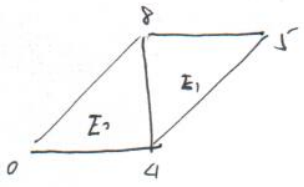
Hence the result is $\left(V_{f_0}^{(0)} \cdot V_{g_3}^{(1)} + V_{f_0}^{(1)} \cdot V_{g_3}^{(2)} + V_{f_0}^{(2)} \cdot V_{g_3}^{(0)}\right) \cdot \frac{1}{3}$

↓

quadrature weight.

Example of 2D

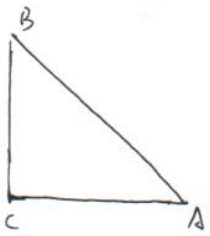
Take edge e_1 $[4, 5]$ as an example



Z_0: $[5, 4, 0]$

Z_1: $[4, 5, 5]$

They all map to



master cell $[A, B, C]$

So for Z_0: $[5, 4] \longrightarrow [A, B]$

for Z_1: $[4, 5] \longrightarrow [A, B]$

Suppose we are calculating $\frac{\nu}{|e_h|}\int_{e_1} \phi_0^{(Z_0)} \cdot \phi_1^{(E_1)} ds$. what should we do?

① choose the quadrature points and get the corresponding values.

Here. $\phi_0^{(Z_0)} \cdot \phi_1^{(Z_1)}$ is a degree 2 function.

According to quadrature method, we choose $t_1 = -\frac{\sqrt{3}}{6} + \frac{1}{2}$ and $t_2 = \frac{\sqrt{3}}{6} + \frac{1}{2}$ as quadrature pts.

we get $\hat{\phi}_0(t_1)$, $\hat{\phi}_0(t_2)$, $\hat{\phi}_1(t_1)$, $\hat{\phi}_2(t_2)$.

$\hat{}$ the value of .

① find the match case

for 2D, there are only 2 case of match

case index	0	1
0		0 1
1	1 0	

We can see in this case

	node 0	node 1
Z 0	8	4
Z 1	4	8

So the match case is $(0,1) \longrightarrow (1,0)$, i.e. case 1

③ get the answer.

Then we can have $\frac{V}{|e_n|}\int_{e_1} \phi_0^{(z_0)} \cdot \phi_1^{(z_1)}\, ds$

$$= \frac{V}{2}\cdot \hat{\phi}_0(t_1)\cdot \hat{\phi}_1(t_2) + \frac{V}{2}\cdot \hat{\phi}_0(t_2)\cdot \hat{\phi}_1(t_1)$$

($|e_n|$ are deleted because $\int_{e_1} \phi_0^{(z_0)} \cdot \phi_1^{(z_1)} = \frac{|e_n|}{1}\cdot \int_{e_1} \hat{\phi}_0^{(z_0)} \cdot \hat{\phi}_1^{(z_1)}\, d\hat{s}$

($\frac{1}{2}$ is the weight of quadrature point)

if we want to calculate the jump term, for example:

$$\int_{e_1} \nabla \phi_0^{(2_0)} \cdot \vec{n}\, \phi_1^{(2_1)}\, ds,$$

Then actually it is simpler than the penalty term, since it has a lower degree.

$$\int_{e_1} \nabla \phi_0^{(2_0)} \cdot \vec{n} \phi_1^{(2_1)}\, ds = \frac{1}{1} \cdot \int_0^1 \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix} \cdot \widehat{\nabla \phi_0} \cdot \vec{n} \cdot \widehat{\phi_1}\, dx$$

$$= \frac{1}{2} \cdot \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix} \cdot \widehat{\nabla \phi_0} \cdot \vec{n} \cdot \widehat{\phi_1}(t).$$

where t is $\frac{1}{2}$. which is the quadrature point, with weight 1.

$\vec{n}$ is the normal vector, in this case is $(1, 0)$

$\widehat{\nabla \phi_0} = (1, 0)^T$

$\phi_1(t) = 1 - \frac{1}{2} = \frac{1}{2}$

So $\int_{e_1} \nabla \phi_0^{(20)} \cdot \vec{n} \cdot \phi_1^{(2_1)}\, ds = \frac{1}{1} \cdot \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot (1, 0) \cdot \frac{1}{2} = 0$

REFERENCE

[1] Sven Berger: Introduction to discontinuous Galerkin element methods, 1, Institute of Aerodynamics, 2003.

[2] Rashid Mehmood, Jon Crowcroft: Parallel iterative solution method for large sparse linear equation systems, 2.1, University of Cambridge, UCAM-CL-TR-650 ISSN 1476-2986, 2005.

[3] Rashid Mehmood, Jon Crowcroft: Parallel iterative solution method for large sparse linear equation systems, 4, University of Cambridge, UCAM-CL-TR-650 ISSN 1476-2986, 2005.

[4] Rashid Mehmood, Jon Crowcroft: Parallel iterative solution method for large sparse linear equation systems, 2.4, University of Cambridge, UCAM-CL-TR-650 ISSN 1476-2986, 2005.

[5] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June M. Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 3.1.1, the Society of Industrial and Applied Mathematics, 2nd edition.

[6] Marzio Sala, Michael A. Heroux, David M. Day, James M. Willenbring: Trilinos Tutorial, 11.4, Sandia Report SAND2004-2189, 2010.

ACKNOWLEDGEMENT

This project was sponsored by Oak Ridge National Laboratory, Joint Institute for Computational Sciences, University of Tennessee, Knoxville and The Chinese University of Hong Kong. It is my great honor to be financially supported by the aforementioned institutions to work on this project in the past ten weeks.

A sincere thank to my mentors, Dr. Ohannes Karakashian, Dr. Kwai Wong and Michael Wisw, for their continuous support and guidance throughout the project. Dr. Karakashian has been not only more than generous in offering intellectual lectures that are interesting and useful. Dr. Wong has been very considerate and full of wisdom in providing prospective insights to problem solving both at work and in daily life. Micheal Wise has helped a lot in some very detail part of the project and make things much easier.

Lastly, thanks for CSURE partcipants, who has worked with me during the two months and been so cheerful and helpful. The memory of this summer will be unforgetable.