第二节 有限元法的常见实现步骤 – 微分方程数值解

笔记本：	我的第一个笔记本
创建时间：	2017/6/12 12:24
URL：	https://iclass.bupt.edu.cn/webapps/blackboard/content/listContent.jsp?course_id=_11052_1&content_id=_52798_1

第二节 有限元法的常见实现步骤

学习指导：F 有限元法 第二节

本节介绍有限元法的常见实现步骤。

作业&思考：F 有限元法 第二节

讲义：F 有限元法 第二节

§8.1 有限元法的常见实现步骤---实际程序设计流程

以求解两点边值问题

$$\begin{cases} Lu = -\dfrac{d}{dx}(p\dfrac{du}{dx}) + qu = f, \quad a < x < b \\ u(a) = 0, \quad u'(b) = 0 \end{cases}$$

的**线性元**为例。

等价变分问题（基于**虚功原理**）：

求$u \in H_E^1(I)$，使

$$a(u,v) = (f,v), \ \forall v \in H_E^1(I)$$

其中

$$\begin{cases} a(u,v) = \displaystyle\int_a^b \left[p\frac{du}{dx} \cdot \frac{dv}{dx} + quv \right] dx \\ (f,v) = \displaystyle\int_a^b fv \ dx \end{cases}$$

步骤：

Step 1 网格生成

Step 2 单元形状函数的构造

Step 3 单元刚度矩阵和单元荷载向量的形成

Step 4 总刚度矩阵和总荷载向量的形成—有限元方程组的形成

Step 5 求解有限元方程组

关于 Step 1

给定$I=[a,b]$的一任意剖分：

节点集合：$a = x_0 < x_1 < \cdots < x_n = b$

第 i 个剖分单元：$e_i = [x_{i-1}, x_i]$

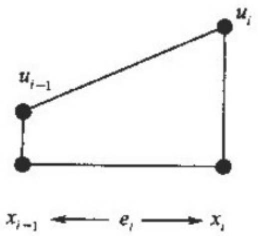
剖分步长：$h_i = x_i - x_{i-1}$

Step 2--Step 4 依赖具体的有限元，以**线性元**为例。

基本作法:

1. 将单元形状函数用**标准单元**中的 Lagrange 因子表示。

单元形状函数 $u^{e_i}(x)$ 图像



为此，首先在**标准单元** $e = [0,1]$ 上定义如下 Lagrange 因子:

$N_0(\xi), N_1(\xi) \in P_1(e)$

$$N_0(0) = 1, N_0(1) = 0 \qquad (2.1)$$

$$N_1(0) = 0, N_1(1) = 1 \qquad (2.2)$$

易知

$$N_0(\xi) = 1 - \xi, N_1(\xi) = \xi$$

图像

节点集合：$a = x_0 < x_1 < \cdots < x_n = b$

第 i 个剖分单元：$e_i = [x_{i-1}, x_i]$

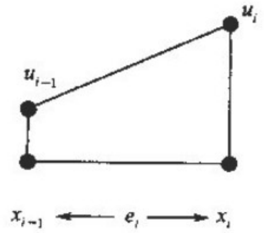
剖分步长：$h_i = x_i - x_{i-1}$

Step 2--Step 4 依赖具体的有限元，以**线性元**为例。

基本作法：

1. 将单元形状函数用**标准单元**中的 Lagrange 因子表示。

单元形状函数 $u^{e_i}(x)$ 图像



为此，首先在**标准单元** $e=[0,1]$ 上定义如下 Lagrange 因子：

$N_0(\xi), N_1(\xi) \in P_1(e)$

$N_0(0)=1, N_0(1)=0 \qquad (2.1)$

$N_1(0)=0, N_1(1)=1 \qquad (2.2)$

易知

$$N_0(\xi)=1-\xi, N_1(\xi)=\xi$$

图像

的**一般项** $a(u_h, v_h)_{e_k}$.

当 $x \in e_k$ 时

$$\begin{cases} u_h(x)=u_{k-1}N_0(\xi)+u_kN_1(\xi) \\ v_h(x)=v_{k-1}N_0(\xi)+v_kN_1(\xi) \end{cases}$$

其中，x 与 ξ 的关系式由 (2.4) 所确定。

$\Rightarrow$

$$
\begin{aligned}
a(u_h,v_h)_{e_k} &= a(u_{k-1}N_0+u_kN_1, v_{k-1}N_0+v_kN_1)_{e_k} \\
&= a(N_0,N_0)_{e_k}u_{k-1}v_{k-1}+a(N_1,N_0)_{e_k}u_kv_{k-1} \\
&\quad +a(N_0,N_1)_{e_k}u_{k-1}v_k+a(N_1,N_1)_{e_k}u_kv_k \\
&= (v_{k-1},v_k)A^{e_k}\begin{pmatrix}u_{k-1}\\ u_k\end{pmatrix}
\end{aligned}
$$

其中

$$
A^{e_k}:=\begin{bmatrix}a_{11}^{e_k} & a_{12}^{e_k}\\ a_{21}^{e_k} & a_{22}^{e_k}\end{bmatrix}=\begin{bmatrix}a(N_0,N_0)_{e_k} & a(N_1,N_0)_{e_k}\\ a(N_0,N_1)_{e_k} & a(N_1,N_1)_{e_k}\end{bmatrix}
$$

称为**单元刚度矩阵**。

类似，考虑虚功方程中的右端

$$
(f,v_h)=\sum_{k=1}^{n}(f,v_h)_{e_k}
$$

的**一般项**。

$$
\begin{aligned}
(f,v_h)_{e_k} &= (f,v_{k-1}N_0+v_kN_1) \\
&= (f,N_0)_{e_k}v_{k-1}+(f,N_1)_{e_k}v_k \\
&= \left(v_{k-1},v_k\right)b^{e_k}
\end{aligned}
$$

其中

$$
b^{e_k}=\begin{pmatrix}b_1^{e_k}\\ b_2^{e_k}\end{pmatrix}=\begin{pmatrix}(f,N_0)_{e_k}\\ (f,N_1)_{e_k}\end{pmatrix}
$$

称为**单元荷载向量**。

3. 总刚度矩阵和总荷载向量

线性元方程组的第 i 个方程的整体表示式为

$$
a(u_h,\phi_i)=(f,\phi_i) \tag{2.5}
$$

其中

$$
u_h=\sum_{j=1}^{n}u_j\phi_j \tag{2.6}
$$

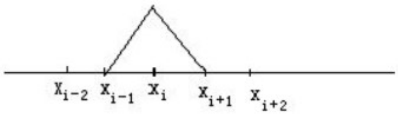
将(2.6)代入(2.5)，并利用 $a(\cdot,\cdot)$ 的双线性性，可得

$$
\sum_{j=1}^{n}a(\phi_j,\phi_i)u_j=(f,\phi_i),i=1(1)n \tag{2.7}
$$

称方程组(2.7)的系数矩阵为**总刚度矩阵**，右端向量为**总荷载向量**。

问量。

注：第 i 个节点基函数 ϕ_i 的图形为



所以，由节点基函数的支集性质知，总刚度矩阵的第 i 行的非零元素由（2.5）中的未知量 u_{i-1},u_i,u_{i+1} 前的系数 $a(\phi_j,\phi_i)$，$j=i-1,i,i+1$ 构成。

4. 总刚度矩阵与单元刚度矩阵的**联系**

主要解决任一单元 e_i 的单元刚度矩阵 A^{e_i} 的元素是如何贡献到总刚度矩阵中的。

建立总刚度矩阵和单元刚度矩阵的联系的基本思想由以下两步构成：

（1）考察第 i 个单元刚度矩阵 A^{e_i} 的元素是出现在线性元方程组的哪些方程中。

首先考察线性元方程组的第 i 个方程的左端，类似于上述引入单元刚度矩阵时的推导过程，我们有

$$a(u_h,\phi_i)=\sum_{k=1}^{n}a(u_h,\phi_i)_{e_k}$$

$$=\sum_{k=1}^{n}((\phi_i)_{k-1},(\phi_i)_k)A^{e_k}\begin{pmatrix}u_{k-1}\\u_k\end{pmatrix}$$

其中 $(\phi_i)_j:=\phi_i(x_j)=\delta_{i,j}$，因此

$$\begin{aligned}a(u_h,\phi_i)&=((\phi_i)_{i-1},(\phi_i)_i)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}+((\phi_i)_i,(\phi_i)_{i+1})A^{e_{i-1}}\begin{pmatrix}u_i\\u_{i+1}\end{pmatrix}\\&=(0\ ,\ 1)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}+(1\ ,\ 0)A^{e_{i+1}}\begin{pmatrix}u_i\\u_{i+1}\end{pmatrix}\end{aligned}\qquad(2.8)$$

再考察线性元方程组的第 $i-1$ 个方程的左端，有

$$\begin{aligned}a(u_h,\phi_{i-1})&=((\phi_{i-1})_{i-2},(\phi_{i-1})_{i-1})A^{e_{i-1}}\begin{pmatrix}u_{i-2}\\u_{i-1}\end{pmatrix}\\&\quad+((\phi_{i-1})_{i-1},(\phi_{i-1})_i)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}\end{aligned}\qquad(2.9)$$

$$=(0\ ,\ 1)A^{e_{i-1}}\begin{pmatrix}u_{i-2}\\u_{i-1}\end{pmatrix}+(1\ ,\ 0)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}$$

类似可考察第 j$(\neq i-1,i)$个线性元方程的左端，易知它们不包含单元刚度矩阵A^{e_i}。

综上，可以得到如下结论：

单元刚度矩阵A^{e_i}仅会出现在线性元方程组的第$i-1$和i个方程中。

这两个方程的特征：它们是第 i 个单元e_i所含的（插值）节点上的自由度（这里为u_{i-1}和u_i）的下标所对应的有限元方程。

(2) 考察在上述方程(即第$i-1$和i个方程)中，单元刚度矩阵A^{e_i}的元素为哪些自由度前的系数。由此可知：它们是如何贡献到总刚度矩阵的相应元素上的。

分别考察(2.8)和(2.9)中与A^{e_i}有关的项。

(a) (2.8)中与A^{e_i}有关的项 (它出现在第 i 个线性元方程中)：

$$(0\ ,\ 1)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}=a_{21}^{e_i}u_{i-1}+a_{22}^{e_i}u_i$$

可知：第 2 行元素$a_{21}^{e_i}$，$a_{22}^{e_i}$出现在第i个方程中，且为第$i-1$和第i个自山度前的系数。

(b) (2.9)中与A^{e_i}有关的项（它出现在第$i-1$个线性元方程中)：

$$(1,\ 0)A^{e_i}\begin{pmatrix}u_{i-1}\\u_i\end{pmatrix}=a_{11}^{e_i}u_{i-1}+a_{12}^{e_i}u_i \qquad (2.11)$$

可知：第 1 行元素$a_{11}^{e_i}$，$a_{12}^{e_i}$出现在第$i-1$个方程中，且为第$i-1$和第i个自由度前的系数；

由上述分析可得结论：关于第i个单元刚度矩阵

$$A^{e_k}=\begin{bmatrix}a_{11}^{e_k} & a_{12}^{e_k}\\a_{21}^{e_k} & a_{22}^{e_k}\end{bmatrix}$$

其第 1 行的元素$a_{11}^{e_i}$，$a_{12}^{e_i}$分别贡献（或叠加）到总刚度矩阵A的第$i-1$行的第$i-1$列和第i列中。

其第 2 行的元素 $a_{21}^{e_i}$ ，$a_{22}^{e_i}$ 分别贡献（或叠加）到总刚度矩阵 A 的第 i 行的第 $i-1$ 列和第 i 列中。即

$$\begin{bmatrix} a_{11}^{e_i} & a_{12}^{e_i} \\ a_{21}^{e_i} & a_{22}^{e_i} \end{bmatrix} \to \begin{bmatrix} & \vdots & \vdots & \\ \cdots & a_{i-1,i-1} & a_{i-1,i} & \cdots \\ \cdots & a_{i,i-1} & a_{i,i} & \cdots \\ & \vdots & \vdots & \end{bmatrix}$$

类似分析可知：2 维单元荷载向量 b^{e_i} 的元素分别贡献到总荷载向量 b 的第 $i-1$ 行和第 i 行中，即

$$\begin{pmatrix} b_1^{e_i} \\ b_2^{e_i} \end{pmatrix} \to \begin{bmatrix} \vdots \\ b_{i-1} \\ b_i \\ \vdots \end{bmatrix}$$

注：上述集成所得到的一定就是线性元方程的总刚度矩阵和总荷载向量。其原因是，我们在集成过程中，没有丢失任何总刚度矩阵和总荷载向量中应包含的信息（因为推导过程均为等式）。为了便于理解，我们打这样一个比方：将一个大盒子中的糖果（可以视为总刚度矩阵和总荷载向量的所有信息）先分别装到 n 个小盒子（可以视为总刚度矩阵和总荷载向量的部分信息）中，然后按某种规则将 n 个小盒子中的糖果取出来，放回大盒子中，只要放回的过程中没有丢失任何糖果，那么最后大盒子中的糖果应该还是它原来里边的糖果。

这样，对于 I 的一任意给定的网格剖分（注意：为了方便 Fortran 语言编程，节点下标从 1 开始）

$$a = x_1 < x_2 < \cdots < x_{n+1} = b$$

上述网格信息分别被存储表示为：

（1） $nnm = n+1$：网格节点数；网格节点（对高次元，还涉及其它类型的节点，如边中点等）坐标

其它类型的节点，如边中点等）坐标：

$$x(i),\ i = 1, \cdots, nnm.$$

（2） $nem = n$：网格单元数；各个单元中的节点的局部编号到其相应的总体编号的映射：

$$node(i,k),\ i = 1,2;\ \mathrm{k} = 1, \cdots, nem,$$

其中

$$node(1,k) = k,\ node(2,k) = k+1.$$

（3）

下面给出生成总刚度矩阵 A 和总荷载向量 b 的算法描述(类语言)。

Step 1 将 A 和 b 初始化为零；

```
Do  i = 1, nnm
        b(i) := 0.0d0
    Do  j = 1, nnm
        a(i, j) := 0.0d0
    End Do
End Do
```

Step 2 生成未经强制（或本质）边界条件处理的总刚度矩阵 A 和总荷载向量 b：

Do $i = 1,\ nem$ 对单元循环

(1) 生成第 i 个单元的单元刚度矩阵

公式为：

$$A^{e_i} = \begin{bmatrix} a(N_0, N_0)_{e_i} & a(N_1, N_0)_{e_i} \\ a(N_0, N_1)_{e_i} & a(N_1, N_1)_{e_i} \end{bmatrix}$$

类语言代码为：

```
Do l=1,2
    Do m=1,2
        A^{e_i}(l,m) = a(N_m, N_l)_{e_i}
    End do
```

End do

End do

(2) 将 A^{e_i} 迭加到总刚度矩阵中

公式为（迭加到 A 的第 i 行和第 $i+1$ 行

的第 i 列和第 $i+1$ 列）：

$$a_{i,i} := a_{i,i} + a(N_0, N_0)_{e_i}$$

$$a_{i,i+1} := a_{i,i+1} + a(N_1, N_0)_{e_i}$$

$$a_{i+1,i} := a_{i+1,i} + a(N_0, N_1)_{e_i}$$

$$a_{i+1,i+1} := a_{i+1,i+1} + a(N_1, N_1)_{e_k}$$

类语言代码为：

Do l=1,2

Do m=1,2

$$A(node(l,i), node(m,i)) = A(node(l,i), node(m,i)) + A^{e_i}(l,m)$$

End do

End do

（3）生成第 i 个单元的单元荷载向量

公式为

$$b^{e_i} = \begin{pmatrix} (f, N_0)_{e_i} \\ (f, N_1)_{e_i} \end{pmatrix}$$

类语言代码为：

Do l=1,2

$$b^{e_i}(l) = (f, N_l)_{e_i}$$

End do

(4)将 b^{e_i} 迭加到总荷载向量 b 中

公式为（迭加到 b 的第 i 行和第 $i+1$ 行中）

$$b_i := b_i + (f, N_0)_{e_i}$$

$$b_{i+1} := b_{i+1} + (f, N_1)_{e_i}$$

类语言代码为：

Do l=1,2

$$b(node(l,i)) = b(node(l,i)) + b^{e_i}(l)$$

```
            End do
      End Do
```

Step 3 生成经强制边界条件处理后的总刚度矩阵 $\tilde{A}$ 和总荷载向量 $\tilde{b}$.

经 Step 2 可得没有经过边界条件处理的 $n+1$ 阶线性有限元方程组（不适定）：

$$AU = b$$

其中

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & \cdots & a_{2,n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & a_{n,n+1} \\ a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} & a_{n+1,n+1} \end{bmatrix}$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_{n+1} \end{pmatrix}, \quad U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ u_{n+1} \end{pmatrix}$$

目标：使得解向量满足**本质边值条件**。

例如：如果要求 $u_1 = \alpha$ ，则可做如下修改

$$A \to \widetilde{A} := \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & \cdots & a_{2,n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n,2} & \cdots & a_{n,n} & a_{n,n+1} \\ 0 & a_{n+1,2} & \cdots & a_{n+1,n} & a_{n+1,n+1} \end{bmatrix}$$

$$b \to \tilde{b} := \begin{pmatrix} \alpha \\ b_2 - a_{21}\alpha \\ \vdots \\ b_n - a_{n,1}\alpha \\ b_{n+1} - a_{n+1,1}\alpha \end{pmatrix}$$

注 1. 关于求解 $n+1$ 阶线性有限元方程组（适定）

$$\tilde{A}U = \tilde{b}$$

的方法。

(1) **直接法**：

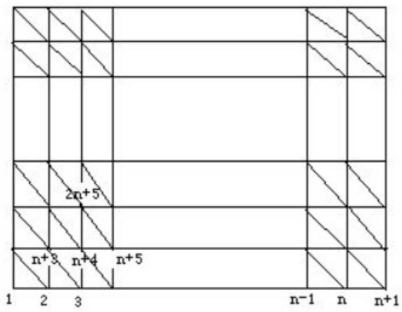
存储量：带状存储＝$O(nb_w)$

运算阶：$O(nb_w^2)$

其中，b_w 为总刚度矩阵的带宽。

$$b_w = \begin{cases} O(1), & for\ \ 1D \\ O(n^{1/2}), & for\ \ 2D \\ O(n^{2/3}), & for\ \ 3D \\ \quad\vdots \end{cases}$$

图示（自然编号）



编号为 $n+4$ 的节点所对应的有限元方程形如：

$$a_{n+4,3}u_3 + a_{n+4,4}u_4 + a_{n+4,n+3}u_{n+3} + a_{n+4,n+4}u_{n+4}$$
$$+a_{n+4,n+5}u_{n+5} + a_{n+4,2n+4}u_{2n+4} + a_{n+4,2n+5}u_{2n+5}$$

适合于**中等以下规模**求解。

(2) **迭代法**：

存储量：稀疏存储$=O(n)$

运算量：对 multigrid 方法：$O(n)$

与并行算法结合，适合于**大规模**求解。

注 2. 关于单元刚度矩阵和单元荷载向量元素的计算。

$$A^{e_k}=\begin{bmatrix} a(N_0,N_0)_{e_k} & a(N_1,N_0)_{e_k} \\ a(N_0,N_1)_{e_k} & a(N_1,N_1)_{e_k} \end{bmatrix}$$

$$b^{e_k}=\begin{pmatrix} (f,N_0)_{e_k} \\ (f,N_1)_{e_k} \end{pmatrix}$$

(1) $a(N_i,N_j)_{e_k}, i,j=0,1$

$$a(N_i,N_j)_{e_k}$$
$$=\int_{e_k}[p(x)N_i'(\xi(x))N_j'(\xi(x))+q(x)N_i(\xi(x))N_j(\xi(x))]dx$$

其中，$\xi:=\xi(x)=\dfrac{x-x_k}{h_k}, x\in e_k$

$\Rightarrow$

$$a(N_i,N_j)_{e_k}$$
$$=h_k\int_0^1[p(x(\xi))N_i'(\xi)N_j'(\xi)+q(x(\xi))N_i(\xi)N_j(\xi)]\,d\xi$$

其中

$$N_i'(\xi)=\frac{dN_i(\xi)}{d\xi}\frac{d\xi}{dx}=h_k^{-1}\frac{dN_i(\xi)}{d\xi}$$

(2) $(f,N_j)_{e_k}, j=0,1$

$$(f,N_j)_{e_k}=\int_{e_k}f(x)N_j(\xi(x))]dx$$
$$=h_k\int_0^1 f(x(\xi))N_j(\xi)d\xi$$

(3) 定义在标准单元$e=[0,1]$上的常见数值积分公式。

$$\int_0^1 g(\xi)d\xi$$

代数精确度

1. 机械求积公式（等距剖分）

$$\int_0^1 g(\xi)d\xi \approx \sum_{k=0}^{m} c_k g(\frac{\kappa}{m})$$

常见：左、右、中矩形公式，梯形，Simpson 公式

2.Gauss 型求积公式

$$\int_0^1 g(\xi)d\xi \approx \sum_{k=0}^{l} \omega_k g(\xi_k)$$

常见：两点 Gauss 求积公式：

$$\int_0^1 g(\xi)d\xi \approx \frac{1}{2}[g(\frac{1}{2}-\frac{\sqrt{3}}{6})+g(\frac{1}{2}+\frac{\sqrt{3}}{6})]$$

◆ **关于高次有限元方法（以二次元为例）**

1. 剖分：

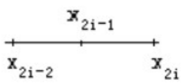
$$a = x_0 < x_2 < \cdots < x_{2(n-1)} < x_{2n} = b$$

$$x_{2j} = jh, j = 0(1)n$$

$$h = \frac{b-a}{n}$$

2. 一般剖分单元和中点：

$$e_i = [x_{2i-2}, x_{2i}], i = 1(1)n$$



3. 自由度的设定，令

$$u_i = u_h(x_i), i = 1(1)2n$$

3. 单元形状函数

$$u^{e_i}(x) = u_{2i-2}N_0(\xi) + u_{2i-1}N_1(\xi) + u_{2i}N_3(\xi)$$

其中变换：

$$\xi := \xi(x) = \frac{x - x_{i-1}}{h_i}$$

将 $e_i \to e = [0,1]$

或等参变换

$$x = x_{2i-2}N_0(\xi) + x_{2i-1}N_1(\xi) + x_{2i}N_3(\xi)$$

将 $e \to e_i$。

实际上，可以验证上述等参变换会退化为仿射变换（其原因是e 为直线段(这时 e 的中点＝其两个端点的平均值)，不是抛物线）：

$$x = h_i\xi + x_{i-1}$$

e 上的 2 次 lagrange 因子：

$$N_0(\xi), N_1(\xi), N_2(\xi) \in P_2(e)$$

满足

$$\begin{cases} N_0(0)=1, N_0(\frac{1}{2})=0, N_0(1)=0 \\ N_1(0)=0, N_1(\frac{1}{2})=1, N_1(1)=0 \\ N_2(0)=0, N_2(\frac{1}{2})=0, N_2(1)=1 \end{cases}$$

其表达式为

$$\begin{cases} N_0(\xi)=2(\xi-1)(\xi-\frac{1}{2}) \\ N_1(\xi)=-4(\xi-1)\xi \\ N_2(\xi)=2\xi(\xi-\frac{1}{2}) \end{cases}$$

4. 单元刚度矩阵（荷载向量）

$$A^{e_k} := \begin{bmatrix} a_{11}^{e_k} & a_{12}^{e_k} & a_{13}^{e_k} \\ a_{21}^{e_k} & a_{22}^{e_k} & a_{23}^{e_k} \\ a_{31}^{e_k} & a_{32}^{e_k} & a_{33}^{e_k} \end{bmatrix} = \begin{bmatrix} a(N_0,N_0)_{e_k} & a(N_0,N_1)_{e_k} & a(N_0,N_2)_{e_k} \\ a(N_1,N_0)_{e_k} & a(N_1,N_1)_{e_k} & a(N_1,N_2)_{e_k} \\ a(N_2,N_0)_{e_k} & a(N_2,N_1)_{e_k} & a(N_2,N_2)_{e_k} \end{bmatrix}$$

$$b^{e_k} = \begin{pmatrix} b_1^{e_k} \\ b_2^{e_k} \\ b_3^{e_k} \end{pmatrix} = \begin{pmatrix} (f,N_0)_{e_k} \\ (f,N_1)_{e_k} \\ (f,N_2)_{e_k} \end{pmatrix}$$

5. 单元刚度矩阵（荷载向量）对总刚度矩阵（荷载向量）的贡献

$$\begin{bmatrix} a_{11}^{e_k} & a_{12}^{e_k} & a_{13}^{e_k} \\ a_{21}^{e_k} & a_{22}^{e_k} & a_{23}^{e_k} \\ a_{31}^{e_k} & a_{32}^{e_k} & a_{33}^{e_k} \end{bmatrix} \to \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & a_{2i-2,2i-2} & a_{2i-2,2i-1} & a_{2i-2,2i} & \cdots \\ \cdots & a_{2i-1,2i-2} & a_{2i-1,2i-1} & a_{2i-1,2i} & \cdots \\ \cdots & a_{2i,2i-2} & a_{2i,2i-1} & a_{2i,2i} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

$$\begin{pmatrix} b_1^{e_k} \\ b_2^{e_k} \\ b_3^{e_k} \end{pmatrix} \to \begin{pmatrix} \vdots \\ b_{2i-2} \\ b_{2i-1} \\ b_{2i} \end{pmatrix}$$

$$\begin{pmatrix} \vdots \end{pmatrix}$$

关于一般高次元可类似进行。