レポート課題1　解説
冬学期講義について

2012年夏学期

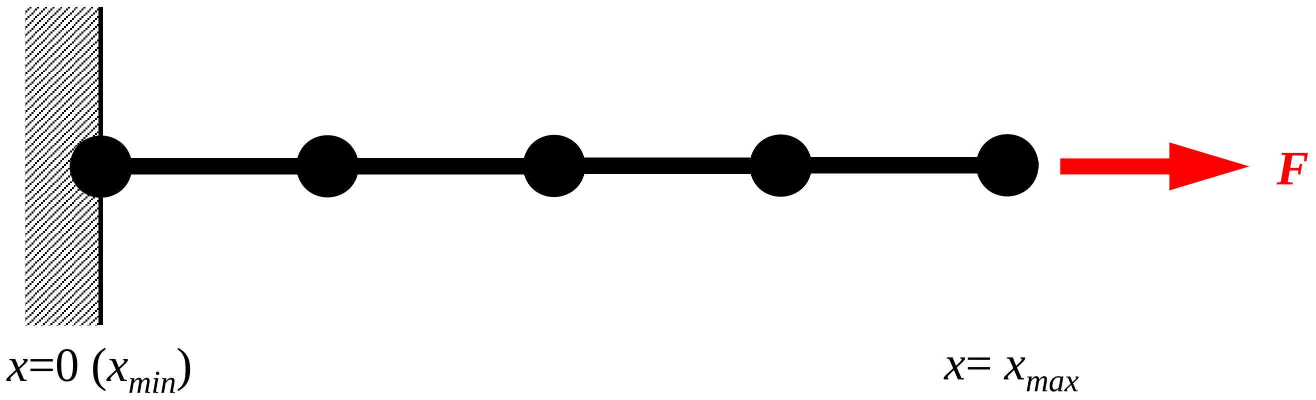
中島 研吾

科学技術計算Ｉ（4820-1027）・コンピュータ科学特別講義Ｉ（4810-1204）

対象とする問題：一次元弾性体（トラス）

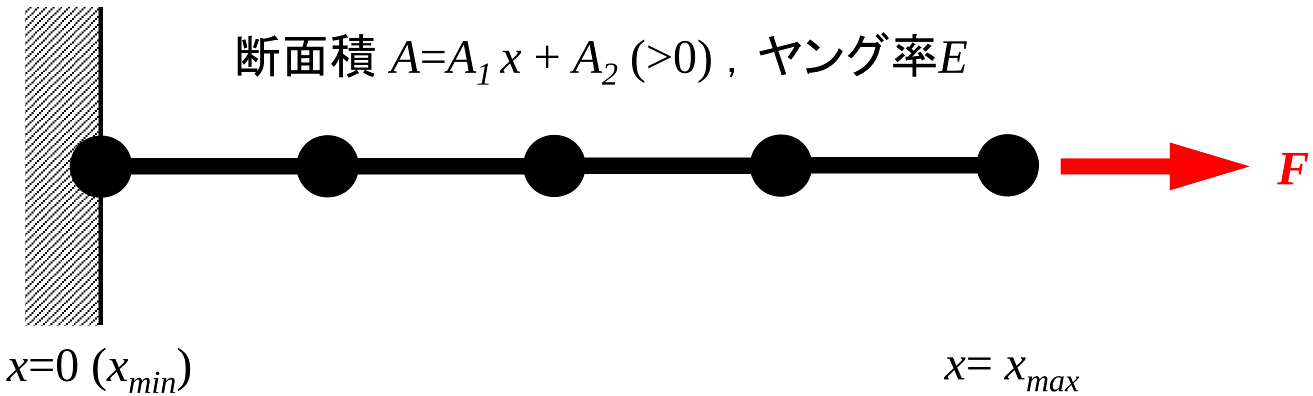
- x方向にのみ自由度（変位u）
  - 一様な：ヤング率E
  - 断面積：$A=A_1 x + A_2$ (>0)
  - 境界条件
    - x=0　：u=0（固定）
    - x=x_{max}：大きさFの力（軸力）
- 自重によるたわみ等はナシ：バネと同じ

対象とする問題：一次元弾性体（トラス）



- x方向にのみ自由度（変位u）
  - 一様な：ヤング率E
  - 断面積：$A=A_1 x + A_2$ (>0)
  - 境界条件
    - $x=0$　：$u=0$（固定）
    - $x=x_{max}$：大きさFの力（軸力）
- 自重によるたわみ等はナシ：バネと同じ

対象とする問題：一次元弾性体（トラス）



つりあい式

$$\frac{\partial \sigma_x}{\partial x} + X = 0$$

ひずみ～変位関係式

$$\varepsilon_x = \frac{\partial u}{\partial x}$$

ひずみ～応力関係式

$$\sigma_x = E\varepsilon_x$$

$$\frac{\partial}{\partial x}\left(E \frac{\partial u}{\partial x} \right) + X = 0$$

変位uを自由度とする支配方程式

計算の手順

- まず変位を求め $$\frac{\partial}{\partial x}\left(E \frac{\partial u}{\partial x} \right) + X = 0$$
- 「ひずみ」を計算し $$\varepsilon_x = \frac{\partial u}{\partial x}$$
- 「応力」を計算する $$\sigma_x = E \varepsilon_x$$

解析解

$$\sigma_x = E\varepsilon_x = \frac{F}{A}$$

$$E\frac{du}{dx} = \frac{F}{A_1 x + A_2}$$

$$Eu = \frac{F}{A_1}\log(A_1 x + A_2) + C \qquad C = -\frac{F}{A_1}\log(A_2) \quad \because u = 0 @ x = 0$$

$$\therefore \quad u = \frac{F}{EA_1}\left[\log(A_1 x + A_2) - \log(A_2)\right]$$

レポート課題1

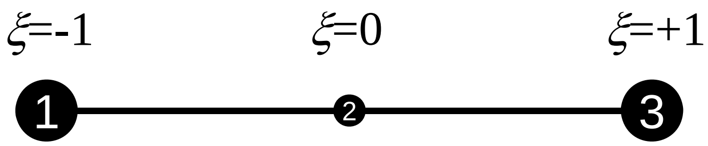
- b1.c，b1.fを高次要素（二次の補間関数使用）の使用により改良せよ（b2.c，b2.f）
- b1とb2についてメッシュ数を変更したケースを実施し，精度について考察せよ
- 下記の成立を確認せよ（断面積一定の場合）：式の展開による

$$\int_V E\left(\frac{d[N]^T}{dx}\frac{d[N]}{dx}\right)dV=\frac{EA}{6L}\begin{bmatrix}+14 & -16 & +2\\ -16 & +32 & -16\\ +2 & -16 & +14\end{bmatrix}$$

- 提出期限
  - 2012年8月24日（金）17:00
- 提出物
  - レポート（概要，計算結果，考察）（図表含A4 5枚以内）
  - プログラムソースリスト

レポート課題1：ヒント①

- 形状関数の微分係数



$$N_1(\xi) = \frac{1}{2}\xi(-1+\xi)$$
$$N_2(\xi) = (1+\xi)(1-\xi)$$
$$N_3(\xi) = \frac{1}{2}\xi(1+\xi)$$

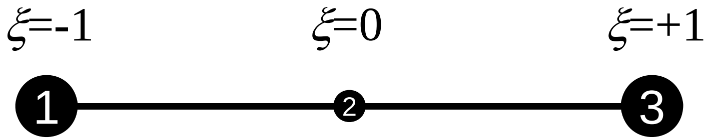
$$\frac{dN_1}{d\xi} = -\frac{1}{2} + \xi$$
$$\frac{dN_2}{d\xi} = -2\xi$$
$$\frac{dN_3}{d\xi} = \frac{1}{2} + \xi$$

$$[Emat] = E\sum_{k=1}^{m}\left[w_k \cdot \frac{1}{|J|}\bigg|_{\xi=\xi_k} \begin{bmatrix} \frac{\partial N_1}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_1}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_1}{\partial \xi}\frac{\partial N_3}{\partial \xi} \\ \frac{\partial N_2}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_2}{\partial \xi}\frac{\partial N_3}{\partial \xi} \\ \frac{\partial N_3}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_3}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi}\frac{\partial N_3}{\partial \xi} \end{bmatrix}\Bigg|_{\xi=\xi_k} A(\xi_k) \right]$$

積分点(ξ_k)における値を代入

レポート課題1：ヒント②

- ヤコビアン



$$\frac{\partial x}{\partial \xi} = \frac{\partial}{\partial \xi}\sum_{i=1}^{3}\left(N_i x_i\right) = \sum_{i=1}^{3}\left(\frac{\partial N_i}{\partial \xi}x_i\right) = \frac{\partial N_1}{\partial \xi}x_1 + \frac{\partial N_2}{\partial \xi}x_2 + \frac{\partial N_3}{\partial \xi}x_3$$

- ガウス積分点（ξ_k）における値

$$\left.\frac{\partial x}{\partial \xi}\right|_{\xi=\xi_k} = \left.\frac{\partial N_1}{\partial \xi}\right|_{\xi=\xi_k}x_1 + \left.\frac{\partial N_2}{\partial \xi}\right|_{\xi=\xi_k}x_2 + \left.\frac{\partial N_3}{\partial \xi}\right|_{\xi=\xi_k}x_3$$

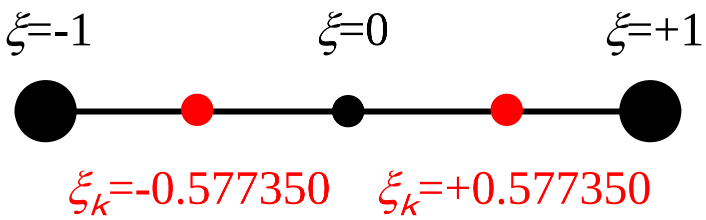
方針

- 一様断面，低次要素（解析解による積分）
  - <$fem1>/1d/1d.c
- 一様断面，高次要素
  - <$fem1>/1d/1d2.c
- 断面変化，低次要素（解析解による積分）
  - <$fem1>/1darea/a1.c
- 断面変化，低次要素（アイソパラメトリック要素，数値積分）
  - <$fem1>/1darea/b1.c
- 断面変化，高次要素（アイソパラメトリック要素，数値積分）
  - 1d2.cを改良するのが実は簡単

ガウスの積分公式

- 無次元化された自然座標系 [-1,+1] を対象とする。
- m個の積分点を使用すると（2m-1）次の関数までは近似可能（従って1次，2次の内挿関数（形状関数）を使用するときは，m=2で十分）

$$\int_{-1}^{+1} f(\xi)d\xi = \sum_{k=1}^{m}\left[w_k \cdot f(\xi_k)\right]$$



$$m=1 \quad \xi_k = 0.00,\ w_k = 2.00$$

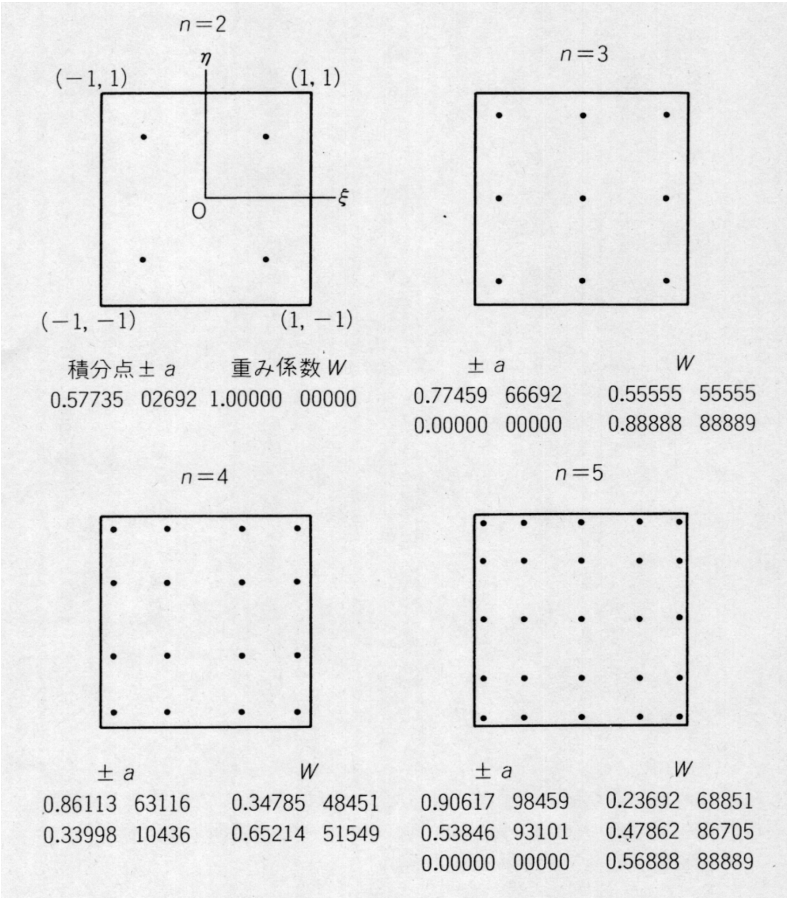
$$m=2 \quad \xi_k = \pm 0.577350,\ w_k = 1.00$$

$$m=3 \quad \xi_k = 0.00,\ w_k = 8/9$$

$$\xi_k = \pm 0.774597,\ w_k = 5/9$$

ガウスの積分公式

二次元，三次元にも容易に拡張可能



$$I = \int_{-1}^{+1}\int_{-1}^{+1} f(\xi,\eta)\, d\xi d\eta$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}\left[W_i \cdot W_j \cdot f(\xi_i,\eta_j)\right]$$

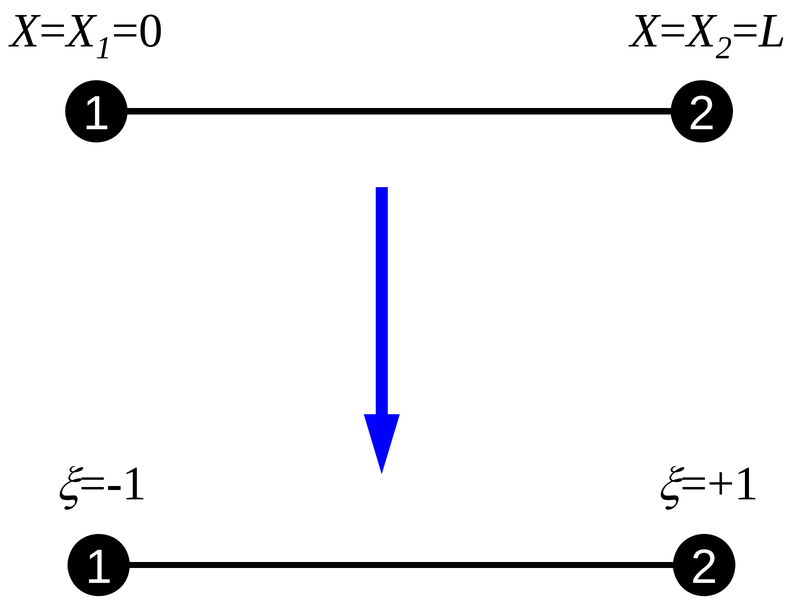
m,n: ξ,η方向の積分点数

(ξ_i,η_j): 積分点の座標値

W_i,W_j: 積分点での重み係数

ガウスの積分公式

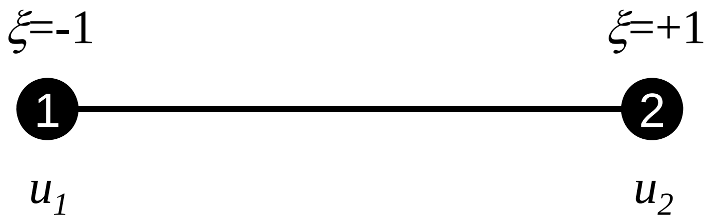
- 使用するためには，$[0,L]$（または$[X_1,X_2]$）から$[-1,+1]$への座標変換が必要
- 内挿関数（形状関数）等も自然座標系上で扱う必要がある



以後，X,u,N などの添字を局所節点番号(1,2,3…)とする

1次（線形）要素

$$u = \alpha_1 + \alpha_2 \xi$$
$$u_1 = \alpha_1 + \alpha_2 (-1)$$
$$u_2 = \alpha_1 + \alpha_2 (+1)$$



$$\alpha_1 = \frac{u_1 + u_2}{2}, \alpha_2 = \frac{-u_1 + u_2}{2}$$

$$u = \alpha_1 + \alpha_2 \xi = \frac{u_1 + u_2}{2} + \frac{-u_1 + u_2}{2} \xi = \underbrace{\frac{1}{2}(1 - \xi)}_{N_1(\xi)} u_1 + \underbrace{\frac{1}{2}(1 + \xi)}_{N_2(\xi)} u_2$$

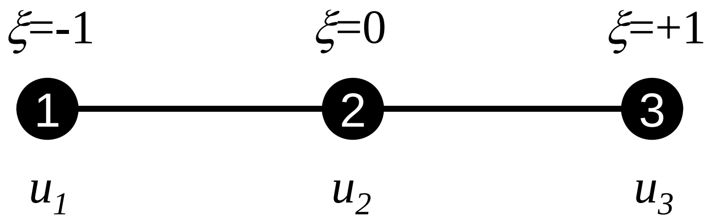
$$N_1(\xi) = \frac{1}{2}(1 - \xi), \quad N_2(\xi) = \frac{1}{2}(1 + \xi)$$

2次要素

$$u=\alpha_1+\alpha_2\xi+\alpha_3\xi^2$$

$$u_1=\alpha_1-\alpha_2+\alpha_3,\quad u_2=\alpha_1$$

$$u_3=\alpha_1+\alpha_2+\alpha_3$$



$$\alpha_1=u_2,\ \alpha_2=\frac{-u_1+u_3}{2},\ \alpha_3=\frac{u_1-2u_2+u_3}{2}$$

$$u=\alpha_1+\alpha_2\xi+\alpha_3\xi^2=u_2+\frac{-u_1+u_3}{2}\xi+\frac{u_1-2u_2+u_3}{2}\xi^2$$

$$=\frac{-\xi+\xi^2}{2}u_1+\left(1-\xi^2\right)u_2+\frac{\xi+\xi^2}{2}u_3$$

$$=\underbrace{\frac{1}{2}\xi\left(-1+\xi\right)}_{N_1(\xi)}u_1+\underbrace{\left(1+\xi\right)\left(1-\xi\right)}_{N_2(\xi)}u_2+\underbrace{\frac{1}{2}\xi\left(1+\xi\right)}_{N_3(\xi)}u_3$$

$$N_1(\xi)=\frac{1}{2}\xi\left(-1+\xi\right),\ N_2(\xi)=\left(1+\xi\right)\left(1-\xi\right),\ N_3(\xi)=\frac{1}{2}\xi\left(1+\xi\right)$$

アイソパラメトリック要素

- 各要素を，自然座標系（local coordinate） [-1,+1]に変換する。
- 各要素の全体座標系（global coordinate） （x）における座標成分を，自然座標系における形状関数[N]を使用して変換する場合，このような要素をアイソパラメトリック要素（isoparametric element）という。

$$u=\sum_{i=1}^{n_N} N_i(\xi)\cdot u_i, \quad x=\sum_{i=1}^{n_N} N_i(\xi)\cdot x_i$$

$$n_N : 2,3...$$

$$u=\frac{1}{2}(1-\xi)u_1+\frac{1}{2}(1+\xi)u_2$$

$$x=\frac{1}{2}(1-\xi)x_1+\frac{1}{2}(1+\xi)x_2$$

$$\left(=\frac{1}{2}(x_2-x_1)\xi+\frac{1}{2}(x_1+x_2)\right)$$

要素単位での積分：$[k]$

$$\int_V E\left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}\right)dV=\int_V E\left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}\right)Adx$$

自然座標系における偏微分（1/2）

- 偏微分の公式より以下のようになる：

$$\frac{\partial N_i(\xi)}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi}$$

$\left[\frac{\partial N_i}{\partial \xi}\right]$ は定義より簡単に求められるが

$\left[\frac{\partial N_i}{\partial x}\right]$ を実際の計算で使用する

$\left[\frac{\partial x}{\partial \xi}\right]$ ヤコビアン（Jacobian）（=J）

$$\frac{\partial x}{\partial \xi} = \frac{\partial}{\partial \xi}\left(\sum_{i=1}^{n_N} N_i x_i\right) = \sum_{i=1}^{n_N} \frac{\partial N_i}{\partial \xi} x_i = J$$

自然座標系における偏微分（2/2）

- 以下のように求められる

$$\frac{\partial N_i}{\partial x} = \frac{\dfrac{\partial N_i(\xi)}{\partial \xi}}{\dfrac{\partial x}{\partial \xi}} = \frac{\partial N_i(\xi)}{\partial \xi} \cdot \frac{1}{J}$$

- 自然座標系における積分

$$\int_0^L f(x)dx = \int_{-1}^{+1} f(\xi)|J|d\xi \quad \because dx = |J|d\xi = \left|\frac{\partial x}{\partial \xi}\right|d\xi$$

要素単位での積分：[k]（1/2）

$$\int_V E\left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}\right)dV = E\int_V \left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}\right)A(x)dx$$

$$= E\int_{-1}^{+1}\left(\frac{\partial[N]^T}{\partial x}\frac{\partial[N]}{\partial x}\right)A(\xi)|J|d\xi = E\int_{-1}^{+1}\left(\left[\frac{\partial[N]^T}{\partial \xi}\frac{1}{J}\right]\left[\frac{\partial[N]}{\partial \xi}\frac{1}{J}\right]\right)A(\xi)|J|d\xi$$

$$= E\int_{-1}^{+1}\left(\frac{1}{|J|}\frac{\partial[N]^T}{\partial \xi}\frac{\partial[N]}{\partial \xi}A(\xi)\right)d\xi$$

$$= E\sum_{k=1}^{m}\left[w_k \cdot \left.\frac{1}{|J|}\right|_{\xi=\xi_k} \left.\frac{\partial[N]^T}{\partial \xi}\right|_{\xi=\xi_k} \left.\frac{\partial[N]}{\partial \xi}\right|_{\xi=\xi_k} A(\xi_k)\right]$$

要素単位での積分：$[k]$（2/2）

$$E\sum_{k=1}^{m}\left[w_k \cdot \frac{1}{|J|}\bigg|_{\xi=\xi_k} \frac{\partial [N]^T}{\partial \xi}\bigg|_{\xi=\xi_k} \frac{\partial [N]}{\partial \xi}\bigg|_{\xi=\xi_k} A(\xi_k) \right]$$

$$= E\sum_{k=1}^{m}\left[w_k \cdot \frac{1}{|J|}\bigg|_{\xi=\xi_k} \begin{bmatrix} \partial N_1 / \partial \xi \\ \partial N_2 / \partial \xi \end{bmatrix} \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi}, \dfrac{\partial N_2}{\partial \xi} \end{bmatrix}\bigg|_{\xi=\xi_k} A(\xi_k) \right]$$

$$= E\sum_{k=1}^{m}\left[w_k \cdot \frac{1}{|J|}\bigg|_{\xi=\xi_k} \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi}\dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_1}{\partial \xi}\dfrac{\partial N_2}{\partial \xi} \\ \dfrac{\partial N_2}{\partial \xi}\dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_2}{\partial \xi}\dfrac{\partial N_2}{\partial \xi} \end{bmatrix}\Bigg|_{\xi=\xi_k} A(\xi_k) \right]$$

解答例(1/7) 諸変数

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>

int main(){
        int NE, N, NPLU, IterMax, errno, NPLU0;
        int R, Z, Q, P, DD, ip;

        double dX, Resid, Eps, Area, F, Young, Jacobi;
        double X1, X2, X3, U1, U2, U3, DL, Strain, Sigma, Ck, XX, X0, A1, A2, DISP;
        double *U, *Rhs, *X;
        double *Diag, *AMat;
        double **W;

        int *Index, *Item, *Icelnod;
        double POI[2], WEI[2], dNdQ[3], Emat[3][3];

        int i, j, in1, in2, in3, k, icel, k12, k13, k21, k23, k31, k32, jS;
        int iter;
        FILE *fp;
        double BNorm2, Rho, Rho1=0.0, C1, Alpha, DNorm2;
        int ierr = 1;
```

解答例(2/7)
初期設定，配列宣言

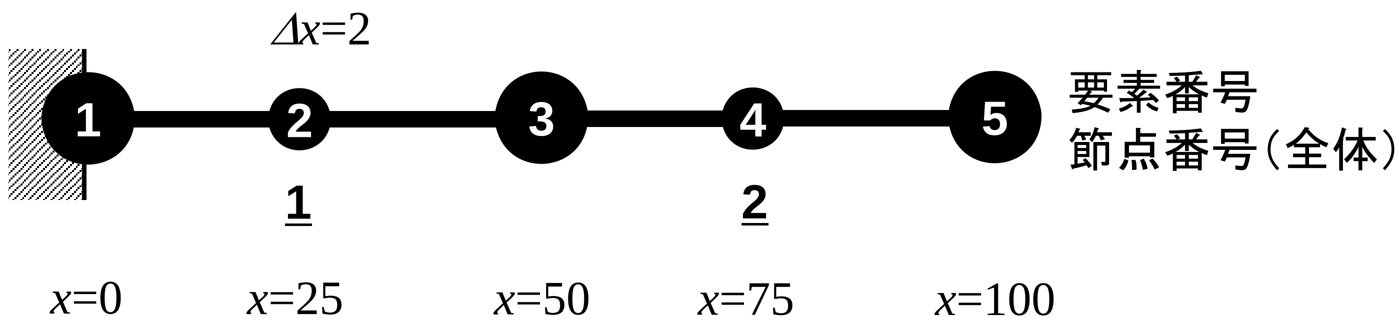
```
/*
// +-------+
// | INIT. |
// +-------+
*/
        fp = fopen("input2.dat", "r");
        assert(fp != NULL);
        fscanf(fp, "%d", &NE);
        fscanf(fp, "%lf %lf %lf %lf %lf", &dX, &F, &A1, &A2, &Young);
        fscanf(fp, "%d", &IterMax);
        fscanf(fp, "%lf", &Eps);
        fclose(fp);

        N     = 2*NE + 1;
        NPLU0= 2*2 + NE*2 + (NE-1)*4;

        U     = calloc(N, sizeof(double));
        X     = calloc(N, sizeof(double));
        Diag = calloc(N, sizeof(double));
        AMat = calloc(NPLU0, sizeof(double));
        Rhs = calloc(N, sizeof(double));
        Index= calloc(N+1, sizeof(int));
        Item = calloc(NPLU0, sizeof(int));
        Icelnod= calloc(3*NE, sizeof(int));
```

入力データ例

制御ファイル input2.dat

2	NE（要素数）
50.0 5.e4 -0.105 12 5.e6	Δx（要素長さL），F，A_1，A_2，E
100	反復回数（CG法後述）
1.e-8	CG法の反復打切誤差

$$x = 0 \quad A_1 x + A_2 = 12.$$
$$x = 100 \quad A_1 x + A_2 = 1.5$$



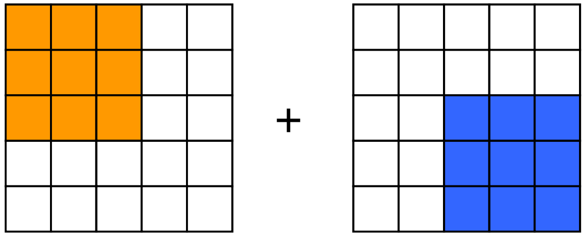
解答例(2/7)
初期設定，配列宣言

```
/*
// +--------+
// | INIT.  |
// +--------+
*/
        fp = fopen("input2.dat", "r");
        assert(fp != NULL);
        fscanf(fp, "%d", &NE);
        fscanf(fp, "%lf %lf %lf %lf %lf", &dX, &F, &A1, &A2, &Young);
        fscanf(fp, "%d", &IterMax);
        fscanf(fp, "%lf", &Eps);
        fclose(fp);

        N     = 2*NE + 1;
        NPLU0= 2*2 + NE*2 + (NE-1)*4;

        U     = calloc(N, sizeof(double));
        X     = calloc(N, sizeof(double));
        Diag = calloc(N, sizeof(double));
        AMat = calloc(NPLU0, sizeof(double));
        Rhs = calloc(N, sizeof(double));
        Index= calloc(N+1, sizeof(int));
        Item = calloc(NPLU0, sizeof(int));
        Icelnod= calloc(3*NE, sizeof(int));
```

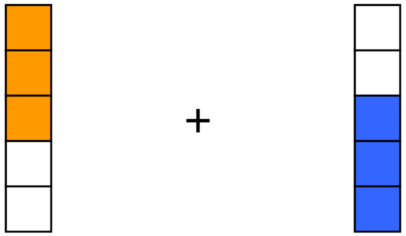
Amat： 非零非対角成分
Item： 対応する列番号

非零非対角成分
同じ要素に属する節点と関係を持つ

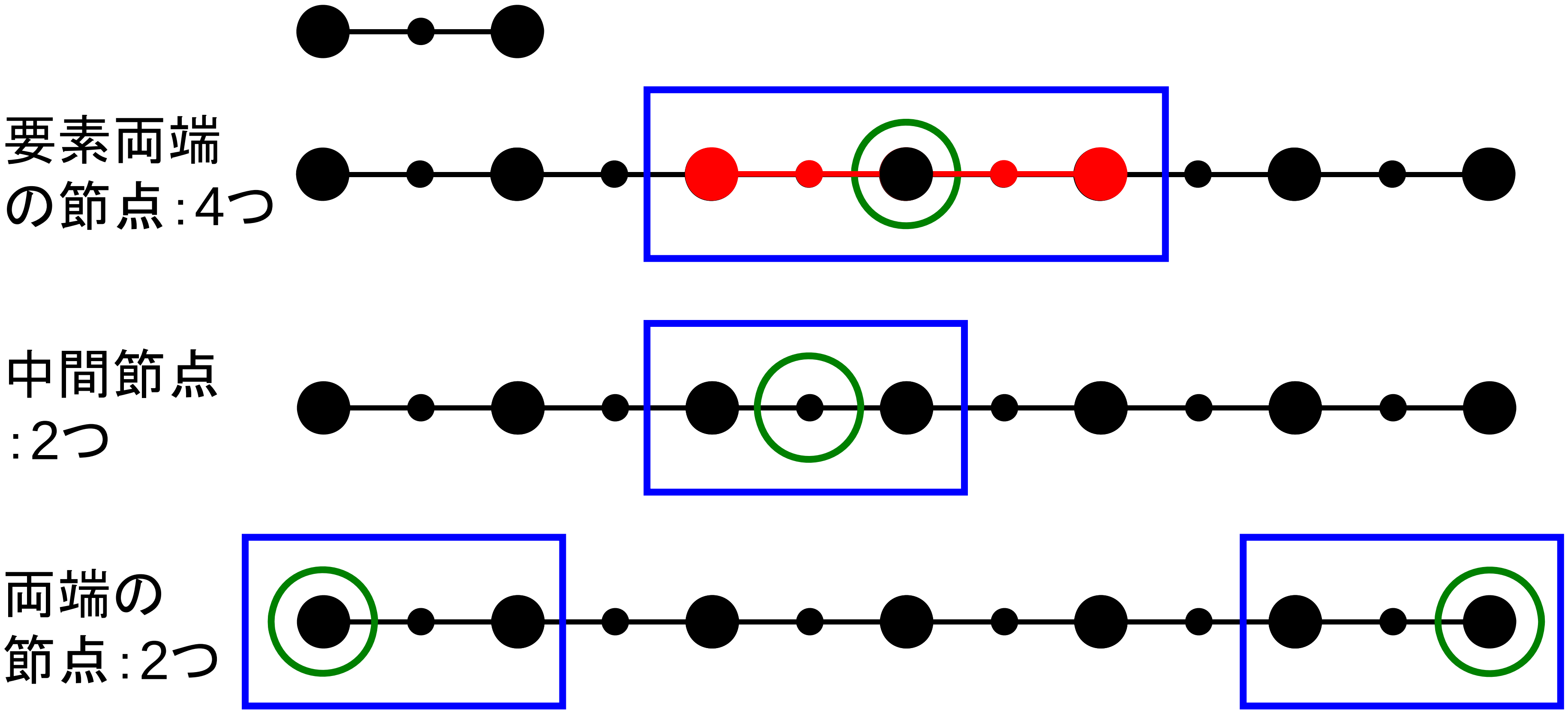
$$[K] = \sum_{i=1}^{2} \left[k^{(i)} \right] =$$



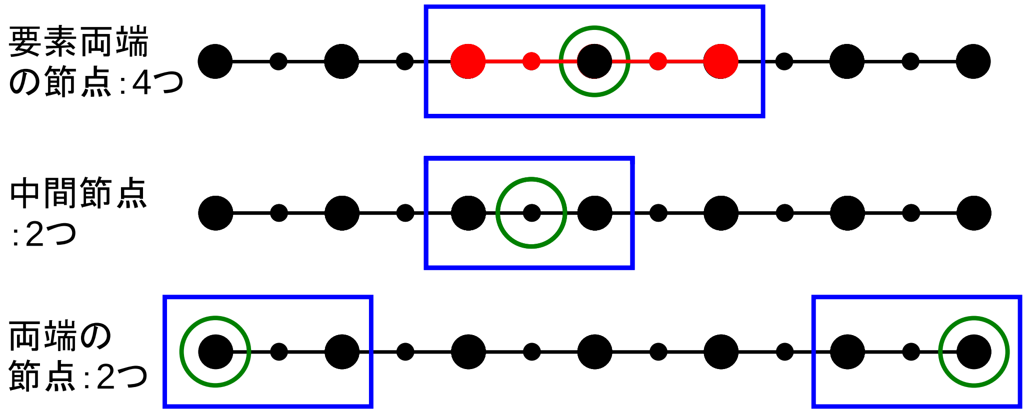
$$\{F\} = \sum_{i=1}^{4} \left\{ f^{(i)} \right\} =$$



非対角成分数：節点によって異なる
同じ要素に属する節点と関係を持つ



非対角成分数：節点によって異なる同じ要素に属する節点と関係を持つ

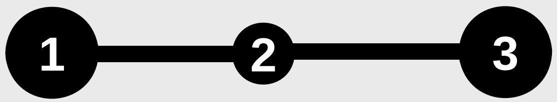


- 「両端」節点数 2
- 中間節点数 NE
- 要素両端節点数 NE+1-2= NE-1
  - （「両端」を除く）
- 従って非零非対角成分数= 2*2 + 2*NE + 4*(NE-1)

解答例(3/7) 初期化(続き)

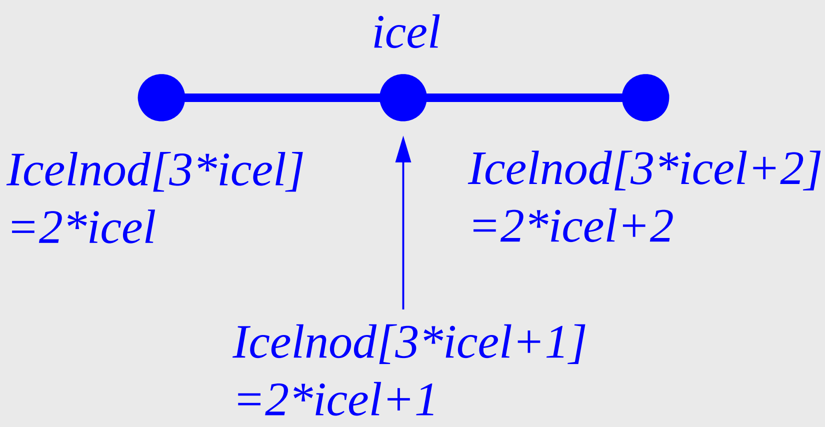
```
W = (double **)malloc(sizeof(double *)*4);
if(W == NULL) {
        fprintf(stderr, "Error: %s¥n", strerror(errno));
        return -1;
}
for(i=0; i<4; i++) {
        W[i] = (double *)malloc(sizeof(double)*N);
        if(W[i] == NULL) {
                fprintf(stderr, "Error: %s¥n", strerror(errno));
                return -1;
        }
}

for(i=0;i<N;i++)    U[i] = 0.0;
for(i=0;i<N;i++) Diag[i] = 0.0;
for(i=0;i<N;i++)  Rhs[i] = 0.0;
for(k=0;k<NPLU0;k++)  AMat[k] = 0.0;
for(i=0;i<N;i++) X[i]= i*dX*0.5;
for(icel=0;icel<NE;icel++){
        Icelnod[3*icel  ]= 2*icel;
        Icelnod[3*icel+1]= 2*icel+1;
        Icelnod[3*icel+2]= 2*icel+2;
}

WEI[0]= +1.0;
WEI[1]= +1.0;
POI[0]= -0.577350;
POI[1]= +0.577350;
```

X：各節点の座標



解答例(3/7) 初期化(続き)

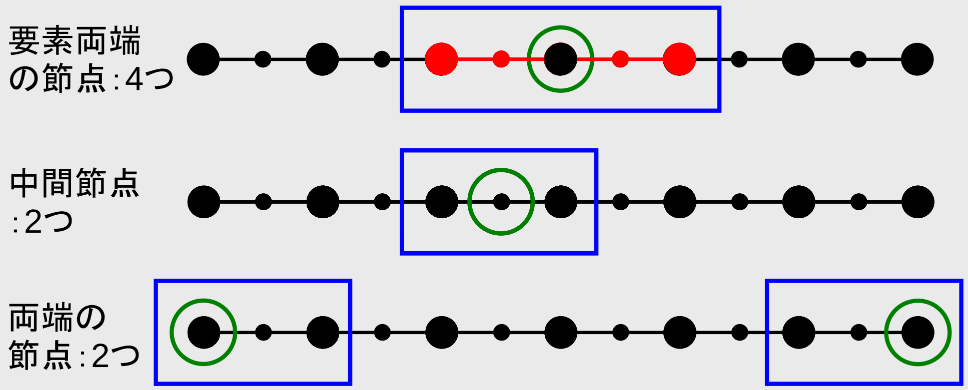
```
W = (double **)malloc(sizeof(double *)*4);
if(W == NULL) {
        fprintf(stderr, "Error: %s¥n", strerror(errno));
        return -1;
}
for(i=0; i<4; i++) {
        W[i] = (double *)malloc(sizeof(double)*N);
        if(W[i] == NULL) {
                fprintf(stderr, "Error: %s¥n", strerror(errno));
                return -1;
        }
}

for(i=0;i<N;i++)    U[i] = 0.0;
for(i=0;i<N;i++) Diag[i] = 0.0;
for(i=0;i<N;i++)  Rhs[i] = 0.0;
for(k=0;k<NPLU0;k++)  AMat[k] = 0.0;
for(i=0;i<N;i++) X[i]= i*dX*0.5;
for(icel=0;icel<NE;icel++){
        Icelnod[3*icel  ]= 2*icel;
        Icelnod[3*icel+1]= 2*icel+1;
        Icelnod[3*icel+2]= 2*icel+2;
}

WEI[0]= +1.0;
WEI[1]= +1.0;
POI[0]= -0.577350;
POI[1]= +0.577350;
```



解答例(4/7)

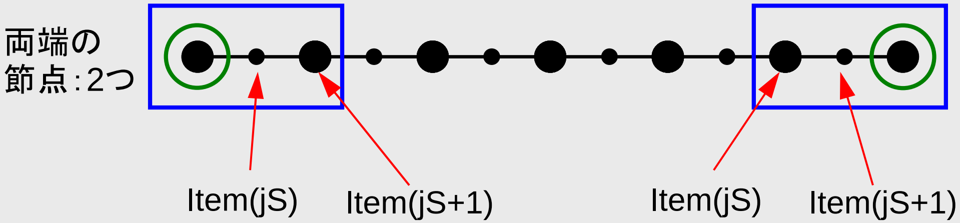
全体マトリクス:非零非対角成分に対応する列番号

```
/*
// +----------------+
// | CONNECTIVITY |
// +----------------+
*/
        Index[0]= 0;
        for(i=1;i<N;i++)  {
          if  (i%2==1)  {Index[i]=4;
          }else          {Index[i]=2;}}
        Index[1]= 2;
        Index[N]= 2;
        for(i=0;i<N;i++){
                Index[i+1]= Index[i+1] + Index[i];}

        NPLU= Index[N];
        for(i=0;i<N;i++){
                int jS = Index[i];
                if(i == 0){
                        Item[jS  ] = i+1;
                        Item[jS+1] = i+2;
                }else if(i == N-1){
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                }else{
                  if (i%2==1){
                        Item[jS  ] = i-1;
                        Item[jS+1] = i+1;
                  } else        {
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                        Item[jS+2] = i+1;
                        Item[jS+3] = i+2;}}}
```



解答例（4/7）
全体マトリクス：非零非対角成分に対応する列番号

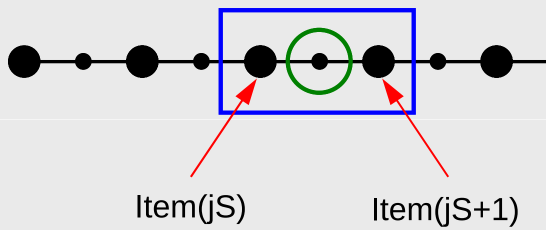
```
/*
// +----------------+
// | CONNECTIVITY |
// +----------------+
*/
        Index[0]= 0;
        for(i=1;i<N;i++) {
          if (i%2==1) {Index[i]=4;
          }else       {Index[i]=2;}}
        Index[1]= 2;
        Index[N]= 2;
        for(i=0;i<N;i++){
                Index[i+1]= Index[i+1] + Index[i];}

        NPLU= Index[N];
        for(i=0;i<N;i++){
                int jS = Index[i];
                if(i == 0){
                        Item[jS  ] = i+1;
                        Item[jS+1] = i+2;
                }else if(i == N-1){
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                }else{
                  if (i%2==1){
                        Item[jS  ] = i-1;
                        Item[jS+1] = i+1;
                  } else       {
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                        Item[jS+2] = i+1;
                        Item[jS+3] = i+2;}}}
```



解答例(4/7)
全体マトリクス：非零非対角成分に対応する列番号

```
/*
// +--------------+
// | CONNECTIVITY |
// +--------------+
*/
      Index[0]= 0;
      for(i=1;i<N;i++) {
        if (i%2==1) {Index[i]=4;
        }else       {Index[i]=2;}}
      Index[1]= 2;
      Index[N]= 2;
      for(i=0;i<N;i++){
              Index[i+1]= Index[i+1] + Index[i];}

      NPLU= Index[N];
      for(i=0;i<N;i++){
              int jS = Index[i];
              if(i == 0){
                      Item[jS  ] = i+1;
                      Item[jS+1] = i+2;
              }else if(i == N-1){
                      Item[jS  ] = i-2;
                      Item[jS+1] = i-1;
              }else{
                if (i%2==1){
                      Item[jS  ] = i-1;
                      Item[jS+1] = i+1;
                } else      {
                      Item[jS  ] = i-2;
                      Item[jS+1] = i-1;
                      Item[jS+2] = i+1;
                      Item[jS+3] = i+2;}}}
```

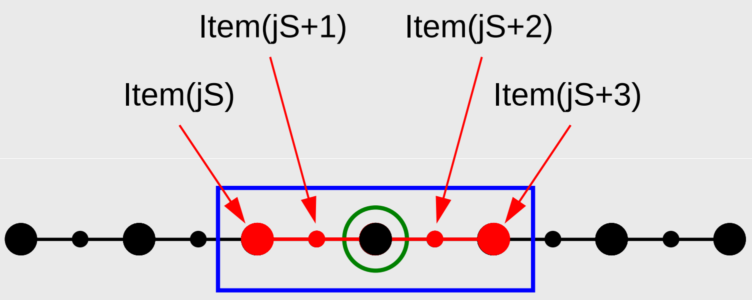
中間節点
:2つ



解答例(4/7)
全体マトリクス：非零非対角成分に対応する列番号

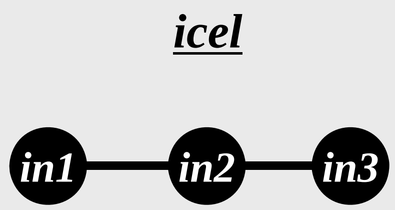
```
/*
// +--------------+
// | CONNECTIVITY |
// +--------------+
*/
        Index[0]= 0;
        for(i=1;i<N;i++) {
          if (i%2==1) {Index[i]=4;
          }else       {Index[i]=2;}}
        Index[1]= 2;
        Index[N]= 2;
        for(i=0;i<N;i++){
                Index[i+1]= Index[i+1] + Index[i];}

        NPLU= Index[N];
        for(i=0;i<N;i++){
                int jS = Index[i];
                if(i == 0){
                        Item[jS  ] = i+1;
                        Item[jS+1] = i+2;
                }else if(i == N-1){
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                }else{
                  if (i%2==1){
                        Item[jS  ] = i-1;
                        Item[jS+1] = i+1;
                  } else       {
                        Item[jS  ] = i-2;
                        Item[jS+1] = i-1;
                        Item[jS+2] = i+1;
                        Item[jS+3] = i+2;}}}
```

要素両端
の節点：4つ



解答例(5/7)　行列生成(1/3)

```
/*
// +-----------------+
// | MATRIX assemble |
// +-----------------+
*/
      for(icel=0;icel<NE;icel++){
        in1= Icelnod[3*icel];
        in2= Icelnod[3*icel+1];
        in3= Icelnod[3*icel+2];
        X1 = X[in1];
        X2 = X[in2];
        X3 = X[in3];

        DL = fabs(X3-X1);
        X0 = 0.5 * (X1+X3);

        Emat[0][0]= 0.0;
        Emat[0][1]= 0.0;
        Emat[0][2]= 0.0;
        Emat[1][0]= 0.0;
        Emat[1][1]= 0.0;
        Emat[1][2]= 0.0;
        Emat[2][0]= 0.0;
        Emat[2][1]= 0.0;
        Emat[2][2]= 0.0;
```

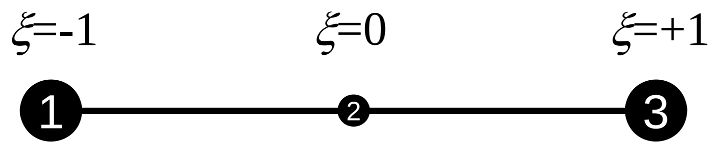


解答例(6/7) 行列生成(2/3)

```
for(ip=0;ip<2;ip++){
  dNdQ[0]= -0.5 + POI[ip];
  dNdQ[1]= -2.0 * POI[ip];
  dNdQ[2]=  0.5 + POI[ip];

  XX= X0 + POI[ip]*0.50*DL;
  Area= A1*XX + A2;

 if(Area<= 0.) {
       fprintf(stderr, "ERROR: Area<0: ¥n");
       return -1;
 }

  Jacobi= fabs(dNdQ[0]*X1 + dNdQ[1]*X2 + dNdQ[2]*X3);
  Ck= Area*Young/Jacobi;
  Emat[0][0]= Emat[0][0] + Ck * WEI[ip] * dNdQ[0] * dNdQ[0];
  Emat[0][1]= Emat[0][1] + Ck * WEI[ip] * dNdQ[0] * dNdQ[1];
  Emat[0][2]= Emat[0][2] + Ck * WEI[ip] * dNdQ[0] * dNdQ[2];
  Emat[1][0]= Emat[1][0] + Ck * WEI[ip] * dNdQ[1] * dNdQ[0];
  Emat[1][1]= Emat[1][1] + Ck * WEI[ip] * dNdQ[1] * dNdQ[1];
  Emat[1][2]= Emat[1][2] + Ck * WEI[ip] * dNdQ[1] * dNdQ[2];
  Emat[2][0]= Emat[2][0] + Ck * WEI[ip] * dNdQ[2] * dNdQ[0];
  Emat[2][1]= Emat[2][1] + Ck * WEI[ip] * dNdQ[2] * dNdQ[1];
  Emat[2][2]= Emat[2][2] + Ck * WEI[ip] * dNdQ[2] * dNdQ[2];
}
```

積分点における形状関数の微分係数

- 形状関数の微分係数



$$N_1(\xi)=\frac{1}{2}\xi(-1+\xi)$$
$$N_2(\xi)=(1+\xi)(1-\xi)$$
$$N_3(\xi)=\frac{1}{2}\xi(1+\xi)$$

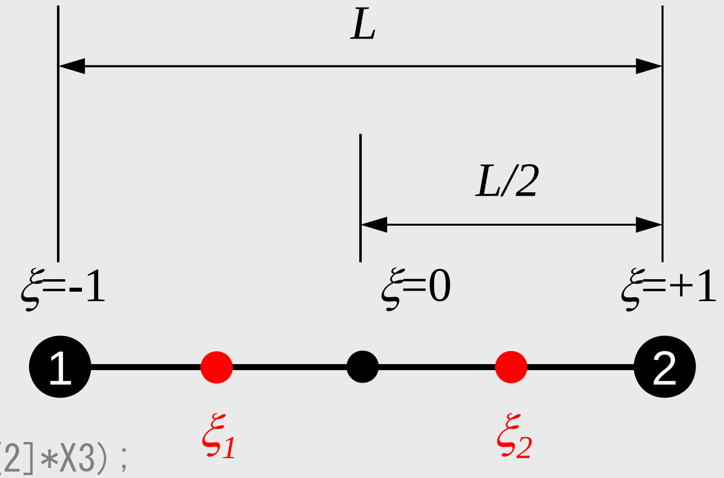
$$\frac{dN_1}{d\xi}=-\frac{1}{2}+\xi$$
$$\frac{dN_2}{d\xi}=-2\xi$$
$$\frac{dN_3}{d\xi}=\frac{1}{2}+\xi$$

$$[Emat]=E\sum_{k=1}^{m}\left[w_k\cdot\frac{1}{|J|}\bigg|_{\xi=\xi_k}\begin{bmatrix}\frac{\partial N_1}{\partial\xi}\frac{\partial N_1}{\partial\xi} & \frac{\partial N_1}{\partial\xi}\frac{\partial N_2}{\partial\xi} & \frac{\partial N_1}{\partial\xi}\frac{\partial N_3}{\partial\xi}\\ \frac{\partial N_2}{\partial\xi}\frac{\partial N_1}{\partial\xi} & \frac{\partial N_2}{\partial\xi}\frac{\partial N_2}{\partial\xi} & \frac{\partial N_2}{\partial\xi}\frac{\partial N_3}{\partial\xi}\\ \frac{\partial N_3}{\partial\xi}\frac{\partial N_1}{\partial\xi} & \frac{\partial N_3}{\partial\xi}\frac{\partial N_2}{\partial\xi} & \frac{\partial N_3}{\partial\xi}\frac{\partial N_3}{\partial\xi}\end{bmatrix}\Bigg|_{\xi=\xi_k} A(\xi_k)\right]$$

積分点(ξ_k)における値を代入

解答例(6/7) 行列生成(2/3)

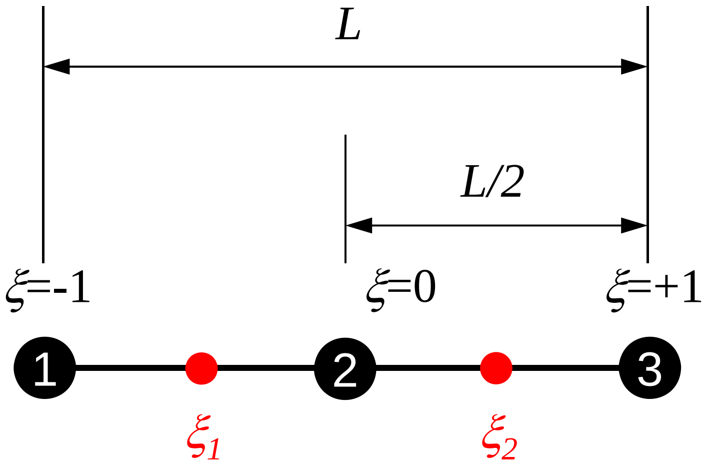
```
for(ip=0;ip<2;ip++){
    dNdQ[0]= -0.5 + POI[ip];
    dNdQ[1]= -2.0 * POI[ip];
    dNdQ[2]=  0.5 + POI[ip];

    XX= X0 + POI[ip]*0.50*DL;
    Area= A1*XX + A2;

   if(Area<= 0.) {
         fprintf(stderr, "ERROR: Area<0: ¥n");
         return -1;
   }

    Jacobi= fabs(dNdQ[0]*X1 + dNdQ[1]*X2 + dNdQ[2]*X3);
    Ck= Area*Young/Jacobi;
    Emat[0][0]= Emat[0][0] + Ck * WEI[ip] * dNdQ
    Emat[0][1]= Emat[0][1] + Ck * WEI[ip] * dNdQ
    Emat[0][2]= Emat[0][2] + Ck * WEI[ip] * dNdQ
    Emat[1][0]= Emat[1][0] + Ck * WEI[ip] * dNdQ[1] * dNdQ[0];
    Emat[1][1]= Emat[1][1] + Ck * WEI[ip] * dNdQ[1] * dNdQ[1];
    Emat[1][2]= Emat[1][2] + Ck * WEI[ip] * dNdQ[1] * dNdQ[2];
    Emat[2][0]= Emat[2][0] + Ck * WEI[ip] * dNdQ[2] * dNdQ[0];
    Emat[2][1]= Emat[2][1] + Ck * WEI[ip] * dNdQ[2] * dNdQ[1];
    Emat[2][2]= Emat[2][2] + Ck * WEI[ip] * dNdQ[2] * dNdQ[2];
}
```



XX　:ガウス積分点における座標
Area: XXにおける断面積

ガウス積分点でのX座標：二次要素

- アイソパラメトリック要素の定義より



$$X(\xi)=\sum_{k=1}^{3}N_k(\xi)X_k$$

$$=N_1(\xi)X_1+N_2(\xi)X_2+N_3(\xi)X_3$$

$$=\frac{1}{2}\xi(-1+\xi)X_1+(1-\xi^2)X_2+\frac{1}{2}\xi(1+\xi)X_3$$

$$=X_2+\frac{X_3-X_1}{2}\xi+\frac{X_1-2X_2+X_3}{2}\xi^2$$

$$=\underline{\frac{X_1+X_3}{2}}+\underline{\frac{X_3-X_1}{2}}\xi \quad \because X_2=\frac{X_1+X_3}{2}$$

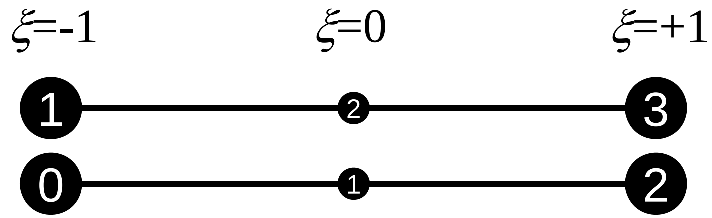
```
∴ XX= X0 + POI[ip]*0.50*DL;
```

解答例(6/7) 行列生成(2/3)

```
for(ip=0;ip<2;ip++){
  dNdQ[0]= -0.5 + POI[ip];
  dNdQ[1]= -2.0 * POI[ip];
  dNdQ[2]=  0.5 + POI[ip];

  XX= X0 + POI[ip]*0.50*DL;
  Area= A1*XX + A2;

 if(Area<= 0.) {
       fprintf(stderr, "ERROR: Area<0: ¥n");
       return -1;
 }

  Jacobi= fabs(dNdQ[0]*X1 + dNdQ[1]*X2 + dNdQ[2]*X3);
  Ck= Area*Young/Jacobi;
  Emat[0][0]= Emat[0][0] + Ck * WEI[ip] * dNdQ[0] * dNdQ[0];
  Emat[0][1]= Emat[0][1] + Ck * WEI[ip] * dNdQ[0] * dNdQ[1];
  Emat[0][2]= Emat[0][2] + Ck * WEI[ip] * dNdQ[0] * dNdQ[2];
  Emat[1][0]= Emat[1][0] + Ck * WEI[ip] * dNdQ[1] * dNdQ[0];
  Emat[1][1]= Emat[1][1] + Ck * WEI[ip] * dNdQ[1] * dNdQ[1];
  Emat[1][2]= Emat[1][2] + Ck * WEI[ip] * dNdQ[1] * dNdQ[2];
  Emat[2][0]= Emat[2][0] + Ck * WEI[ip] * dNdQ[2] * dNdQ[0];
  Emat[2][1]= Emat[2][1] + Ck * WEI[ip] * dNdQ[2] * dNdQ[1];
  Emat[2][2]= Emat[2][2] + Ck * WEI[ip] * dNdQ[2] * dNdQ[2];
}
```

$$
\begin{aligned}
Jacobi &= \left|\frac{\partial x}{\partial \xi}\right| = \left|\frac{\partial}{\partial \xi}\sum_{i=1}^{3}(N_i x_i)\right| \\
&= \left|\sum_{i=1}^{3}\left(\frac{\partial N_i}{\partial \xi}x_i\right)\right| = \left|\frac{\partial N_1}{\partial \xi}x_1 + \frac{\partial N_2}{\partial \xi}x_2 + \frac{\partial N_3}{\partial \xi}x_3\right|
\end{aligned}
$$

積分点におけるヤコビアン



- ヤコビアン

$$\left|\frac{\partial x}{\partial \xi}\right| = \left|\frac{\partial}{\partial \xi}\sum_{i=1}^{3}\left(N_i x_i\right)\right| = \left|\sum_{i=1}^{3}\left(\frac{\partial N_i}{\partial \xi}x_i\right)\right| = \left|\frac{\partial N_1}{\partial \xi}x_1 + \frac{\partial N_2}{\partial \xi}x_2 + \frac{\partial N_3}{\partial \xi}x_3\right|$$

- ガウス積分点（ξ_k）における値

$$\left.\frac{\partial x}{\partial \xi}\right|_{\xi=\xi_k} = \left.\frac{\partial N_1}{\partial \xi}\right|_{\xi=\xi_k} x_1 + \left.\frac{\partial N_2}{\partial \xi}\right|_{\xi=\xi_k} x_2 + \left.\frac{\partial N_3}{\partial \xi}\right|_{\xi=\xi_k} x_3$$

解答例(6/7) 行列生成(2/3)

```
for(ip=0;ip<2;ip++){
  dNdQ[0]= -0.5 + PO
  dNdQ[1]= -2.0 * PO
  dNdQ[2]=  0.5 + PO

  XX= X0 + POI[ip]*0.
  Area= A1*XX + A2;

 if(Area<= 0.) {
      fprintf(stder
      return -1;
 }

  Jacobi= fabs(dNdQ[0]*X1 + dNdQ[1]*X2 + dNdQ[2]*X3);
  Ck= Area*Young/Jacobi;
  Emat[0][0]= Emat[0][0] + Ck * WEI[ip] * dNdQ[0] * dNdQ[0];
  Emat[0][1]= Emat[0][1] + Ck * WEI[ip] * dNdQ[0] * dNdQ[1];
  Emat[0][2]= Emat[0][2] + Ck * WEI[ip] * dNdQ[0] * dNdQ[2];
  Emat[1][0]= Emat[1][0] + Ck * WEI[ip] * dNdQ[1] * dNdQ[0];
  Emat[1][1]= Emat[1][1] + Ck * WEI[ip] * dNdQ[1] * dNdQ[1];
  Emat[1][2]= Emat[1][2] + Ck * WEI[ip] * dNdQ[1] * dNdQ[2];
  Emat[2][0]= Emat[2][0] + Ck * WEI[ip] * dNdQ[2] * dNdQ[0];
  Emat[2][1]= Emat[2][1] + Ck * WEI[ip] * dNdQ[2] * dNdQ[1];
  Emat[2][2]= Emat[2][2] + Ck * WEI[ip] * dNdQ[2] * dNdQ[2];
}
```

$$[Emat] = E\sum_{k=1}^{m}\left[w_k \cdot \frac{1}{|J|}\bigg|_{\xi=\xi_k} \begin{bmatrix} \frac{\partial N_1}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_1}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_1}{\partial \xi}\frac{\partial N_3}{\partial \xi} \\ \frac{\partial N_2}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_2}{\partial \xi}\frac{\partial N_3}{\partial \xi} \\ \frac{\partial N_3}{\partial \xi}\frac{\partial N_1}{\partial \xi} & \frac{\partial N_3}{\partial \xi}\frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi}\frac{\partial N_3}{\partial \xi} \end{bmatrix}\Bigg|_{\xi=\xi_k} A(\xi_k) \right]$$

解答例（7/7） 行列生成（3/3）

1d2.cと同じ

```
        Diag[in1]= Diag[in1] + Emat[0][0];
        Diag[in2]= Diag[in2] + Emat[1][1];
        Diag[in3]= Diag[in3] + Emat[2][2];

        if (icel==0){k12=Index[in1];
                     k13=Index[in1]+1;
              }else {k12=Index[in1]+2;
                     k13=Index[in1]+3;}

          k21=Index[in2];
          k23=Index[in2]+1;

          k31=Index[in3];
          k32=Index[in3]+1;

          AMat[k12]= AMat[k12] + Emat[0][1];
          AMat[k13]= AMat[k13] + Emat[0][2];
          AMat[k21]= AMat[k21] + Emat[1][0];
          AMat[k23]= AMat[k23] + Emat[1][2];
          AMat[k31]= AMat[k31] + Emat[2][0];
          AMat[k32]= AMat[k32] + Emat[2][1];
      }
```

$$
\begin{array}{l}
\text{1行目}: in1\text{の非対角成分} \\
\text{2行目}: in2\text{の非対角成分} \\
\text{3行目}: in3\text{の非対角成分}
\end{array}
\begin{bmatrix}
- & k12 & k13 \\
k21 & - & k23 \\
k31 & k32 & -
\end{bmatrix}
$$

解答例(7/7) 行列生成(3/3)

1d2.cと同じ

```
    Diag[in1]= Diag[in1] + Emat[0][0];
    Diag[in2]= Diag[in2] + Emat[1][1];
    Diag[in3]= Diag[in3] + Emat[2][2];

    if (icel==0){k12=Index[in1];
                 k13=Index[in1]+1;
           }else {k12=Index[in1]+2;
                 k13=Index[in1]+3;}

      k21=Index[in2];
      k23=Index[in2]+1;

      k31=Index[in3];
      k32=Index[in3]+1;

      AMat[k12]= AMat[k12] + Emat[0][1];
      AMat[k13]= AMat[k13] + Emat[0][2];
      AMat[k21]= AMat[k21] + Emat[1][0];
      AMat[k23]= AMat[k23] + Emat[1][2];
      AMat[k31]= AMat[k31] + Emat[2][0];
      AMat[k32]= AMat[k32] + Emat[2][1];
  }
```

$$
\begin{array}{l}
\text{1行目}: in1\text{の非対角成分} \\
\text{2行目}: in2\text{の非対角成分} \\
\text{3行目}: in3\text{の非対角成分}
\end{array}
\begin{bmatrix}
- & k12 & k13 \\
k21 & - & k23 \\
k31 & k32 & -
\end{bmatrix}
$$

冬学期講義

科学技術計算Ⅱ・コンピュータ科学特別講義Ⅱ（並列有限要素法）

- 科学技術計算Ⅰ・コンピュータ科学特別講義Ⅰ（有限要素法）（夏学期）に引き続き以下の講義，実習を実施：
  - MPIによる並列計算プログラミング入門
  - 並列有限要素法のためのデータ構造
  - 並列プログラムの作成法
  - FX10（Oakleaf-FX）によるプログラミング実習

- 夏学期に扱った三次元弾性静解析プログラム「fem3d」の並列化をMPIによって実施する

科学技術計算（有限要素法） I, II

- 担当：中島研吾
- 対象者：大学院
- 開講時期＆単位数
  - I: 夏学期，2単位
  - II: 冬学期，2単位

- 概要
  - Iでは，科学技術シミュレーションで広く使用されている有限要素法の基礎的な理論から実用的なプログラムの作成法まで，弾性静力学を題材として，連立一次方程式解法等周辺技術も含めて講義を実施し，プログラミングの実習を行う。
  - IIでは，並列有限要素法のためのデータ構造，並列プログラムの作成法，「HPC-MW」，「ppOpen-HPC」等の大規模並列シミュレーションコード開発基盤についても講義し，情報基盤センタースパコンによるプログラミング実習を実施する。
  - **大規模並列シミュレーションにおいては，科学・工学と計算機科学・数理科学の専門家の密接な協力が必要である。本講義は，単に並列アプリケーション開発技術を習得するだけでなく，特に情報理工学系研究科の学生がアプリーケーション側のニーズを把握し，両分野の融合領域を開拓する問題意識を育てることを目的としている。**
  - I, IIとも課題のレポート採点で成績をつける

夏学期（I）

1. 有限要素法の基礎理論・弾性静力学
2. ガラーキン法による有限要素法
3. 疎行列解法，前処理手法
4. 有限要素法プログラミング解説
   - 一次元問題
   - 三次元問題
5. プログラミング実習（情報基盤センターECCSシステムを使用）

冬学期（II）

1. MPIによる並列計算プログラミング入門
2. 並列有限要素法のデータ構造
3. 並列有限要素法プログラムの開発
4. 大規模並列シミュレーションコード開発基盤
5. プログラミング実習（情報基盤センタースパコンを使用）