

アイソパラメトリック要素による有限要素法シミュレーション

東京大学大学院工学系研究科機械工学専攻 泉 聡志

平成 15 年 10 月 28 日

目次

1	二次元シミュレータ	2
1.1	モデル	5
1.2	物性値 (弾性定数)	5
1.3	要素	5
1.4	荷重	6
1.5	出力	7
1.6	入力データのチェック	7
1.7	その他	9
2	三次元シミュレータ	13
2.1	要素	14
2.2	Equation 機能の追加	15
3	連立一次方程式の解法	16
3.1	修正コレスキー法による LU 分解 (三次元シミュレータ)	16
3.2	共役勾配法による解法 (三次元シミュレータ)	17
3.3	スカイライン法	17
4	† 弾性定数とヤング率	17
4.1	Orthotropic(Orthorhombic)	18
4.2	Cubic	19
4.3	Isotropic	20
5	†Equation 機能の追加に伴うプログラム修正点 (三次元シミュレータ)	21
6	† 要素内局所座標の算出	22

有限要素法シミュレータ HISAK に修正を加え、2次元・3次元においてアイソパラメトリック要素を扱えるようにしたものである。入力ファイル・出力ファイルには若干の違いがある。言語は FORTRAN77 で書かれている。これはマニュアルではなく、著者の覚書である。

有限要素法の手法の詳細は文献 [1][2] を参考にした。

1 二次元シミュレータ

二次元シミュレータはガウスの消去法のソルバのみが対応している。二次元 FEM のプログラムのフローチャートを図 1 に示す。また、入力ファイルのフォーマットを図 2 に示す。HISAK と比較して若干変更した。以下に、メインプログラムとその注釈を示す。

```

c   main program   with comment by izumi 1998-03-30
c-----
c   fortran77 では、配列のアロケートが出来なかったため、予想される配列の最大
c   の大きさをあらかじめ定義しておき、それ以下の大きさで配列を使わなければ
c   ならなかった。fortran90 ではアロケートが可能になっているので、このような
c   設定は不要となっている。また、サブルーチン内での配列宣言が f(*) など*を
c   使った表記になっているが、これも古い表記であり、*を使うことは推奨されない。
c
c   ielm  : max. number of an element  最大要素数
c   inode : max. number of a node      最大節点数
c   ird   : max. number of integration point in one element  1要素の最大積分点数
c   igk   : max. size of stiffness matrix  inode*idim  剛性マトリックスの最大サイズ
c   ibn   : max. size of half band width  バンド幅の半分の最大サイズ
c   ibound: max. number of constrain (force and displacement)  拘束の最大数
c   ina   : max. number of integration point  ielm*ird  積分点の最大数
c   inmte : max. number of a node in a element --> 8  要素内の最大節点数
c   inmmn : max. size of element stiffness matrix  inmte*idim  要素剛性マトリックスの最大サイズ
c   imnum : max. number of the kind of material  最大の材料の種類の数
c   idm   : dimension  次元
c   icom  : number of stress component (2-dim->4 ,3dim->6)  応力成分の数
c-----
c       include 'para.inc' ! 共有する上記最大数設定数
c       include 'const.inc' ! 共有する定数
c-----
c   dndx1(i): dni/dxi, dndeta(i) : dni/deta  (ni:形状関数)  形状関数の微分
c   x(i) : x-coordinate of node i ,y(i) : y-coordinate  座標
c-----
c       dimension dndxi(inmte),dndeta(inmte),x(inode),y(inode),
c-----
c   b(i,j) : elemental stiffness matrix  bマトリックス
c   axi(i) : x-coordinate of i-th integration point ( -1 < axi < 1 )  ξ
c   aeta(i): y-coordinate of i-th integration point ( -1 < aeta < 1 )  η
c   h(i)   : weight parameter of i-th integration point  ガウス積分の重みパラメータ
c   d(i,j) : elastic matrix (1-x,2-y,3-z,4-xy)  弾性マトリックス
c-----
c       +   b(icom,inmmn),axi(ird),aeta(ird),h(ird),d(icom,icom),
c-----
c   c(i,j) : bt(inmmn,4)*d(4,4)*b(4,inmmn) bt*d*b
c   ek(i,j) : element stiffness matrix  要素剛性マトリックス
c   nn(i,j) : node number of i-th local node of element j  要素 j の i 番目の節点の番号

```

```

-----
+      c(inmnm,inmnm),ek(inmnm,inmnm),nn(inmte,ielm),
-----
c gk(i,j) : global stiffness matrix  i: matrix size  j:half band width  全体剛性マトリックス (マトリックスサイ
ズ×バンド半幅)
c f(i)    : force vector of node i  力ベクトル
c ifor(i) : freedom number of i-th initial load point  i 番目の力をかける自由度番号
c fb(i)   : load value of i-th initial load point  i 番目の力の値
-----
+      gk(igk,ibn),f(igk),ifor(ibound),fb(ibound),
-----
c idis(i) : freedom number of i-th initial disp. point  i 番目の拘束自由度
c ub(i)   : disp. value of i-th initial disp. point  i 番目の拘束の変位値
c u(i)    : displacement value of i-th freedom  i 番目の自由度の変位ベクトル
c ep(i,j,k) : i-th component strain at j-th element
c          k-th local integration point  j 要素、k 積分点におけるひずみの i 番目の成分
c bm(i)    : for strain (b matrix for i-th integration point)  ひずみ
c sg(i,j,k) : i-th component stress at j-th element
c          k-th local integration point  j 要素、k 積分点における応力の i 番目の成分
c bt(i)    : transposed b(4,inmnm) matrix
c btd(i)   : bt(inmnm,4)*d(4,4)
-----
c      +      idis(ibound),ub(ibound),u(igk),ep(icom,ina),
+      idis(ibound),ub(ibound),u(igk),ep(icom,ielm,ird),
+      bm(ina,icom,inmnm),sg(icom,ielm,ird),bt(inmnm,icom),
+      btd(inmnm,icom),
-----
ccc yo(i) : young modulus of i-th material  i 番目の材料のヤング率
ccc po(i) : poisson ratio of i-th material  i 番目の材料のポアソン比
c econ(i,j) : i-th elastic constants of j-th material  j 番目の材料の i 番目の弾性定数
c iecon(i) : elastic character of i-th material  i 番目の材料の弾性的性質
c nom(i) : material number of i-th element  i 番目の材料の材料番号
c nmte(i) : number of a node in i-th element  i 番目の節点数
c ngng(i) : number of integration point in i-th element  i 番目の材料の積分点数
c neus(i) : i : program node number , neus(i) : user element number
c nnus(i) : i : program node number , nnus(i) : user node number ユーザー要素番号とプログラムの要素 (節点) 番号
との対応
-----
c      +      yo(imnum),po(imnum),nom(ielm),nmte(ielm),ngng(ielm)
+      econ(9,imnum),iecon(imnum),nom(ielm),nmte(ielm),ngng(ielm),
+      neus(ielm),nnus(inode)
-----
c n      : shape function ni          形状関数
c jinv   : inverse of jacobian matrix ヤコビアンマトリックスの逆
c jm     : jacobian matrix          ヤコビアンマトリックス
-----
double precision n(inmte),jinv(idim,idim),jm(idim,idim)
c
open(77,file='datacheck.dat')
open(2,file='result.dat')
-----
c for subrouitie data
c ngauss : order of gauss integration  ガウス積分の次数
c npara  : analysis condition (0:plain stress,1:plain strain)  平面応力・ひずみ場の選択
c t      : thickness  板厚

```

```

c ntn      : total number of nodes      節点数
c nte      : total number of elements   要素数
c ntf      : total number of freedoms   自由度数
c ie       : element number(variables) 要素番号 (変数)
c ntfor    : total number of freedoms with initial load 初期荷重の自由度数
c ntdis    : total number of freedoms with initial displacement 初期変位の自由度数
c nna      : the total number of integration points 積分点数
c mnum     : number of the kind of material 材料の種類
c-----
c      call data(a,nmte,ngauss,npara,econ,t,ntn,nte,ntf,
c      +      nn,x,y,ie,ntfor,ifor,fb,ntdis,idis,ub,ngng,nna,
c      +      nom,mnum,iecon)
c      call data(a,nmte,econ,t,          ! 入力データの読み込み フォーマットはマニュアルを参照
c      +      nn,x,y,ifor,fb,idis,ub,ngng,nom,iecon,neus,nnus)
c--- band width
c-----
c nb       : half band width
c-----
c      call band(nn,nmte) ! バンド幅の計算
c
c      call check ! データの基本的なチェック
c
c      call gauss(axi,aeta,h,nmte,ngauss)
c      call clear(ek,gk,f) ! 配列の初期化 Fortran90 では単に、ek=0.0d0; gk=0.0d0; f=0.0d0 と書けばよい。
c      na=0
ccc for all element  すべての要素について
c      do 150 ie=1,nte
c--- gauss integration point ガウスの積分点の設定 (マニュアル参照)
c      call gauss(axi,aeta,h,nmte)
c--- elastic matrix 弾性マトリックスの設定 (マニュアル参照)
c      call dmat(econ,d,nom,iecon)
ccc for all integration point  すべての積分点について
c      do 100 kg=1,ngng(ie)
c      xi=axi(kg)
c      eta=aeta(kg)
c      hnum=h(kg)
c-----
c na       : through number of integration point
c-----
c      na=na+1
c-----
c rx       : real space x-coordinate of integration point
c      for axial symmetry element
c-----
c--- apply to interpolation function n, dn/dxi, dn/deta 内装関数に  $\xi$ ,  $\eta$  を代入
c      call int(xi,eta,n,dndxi,dndeta,nmte,rx,x,nn)
c--- jacobian matrix and its inverse matrix
c-----
c detj     : det[j]          ヤコビアンマトリックスとその逆を計算
c-----
c      call jmat(x,y,dndxi,dndeta,jm,jinv,detj,nn,nmte)
c b matrix Bマトリックスの計算
c      call bmat(jinv,b,n,dndxi,dndeta,nmte,rx)
c b matrix for each integration point (na) ひずみ計算のための積分点毎のBマトリックスの計算
c      call bmmat(b,na,bm,nmte)

```

```

c btdb matrix [B]^t[D][B]-->c(*,*)
      call bdbmat(b,c,d,nmte,bt,btd)
c element stiffness matrix 要素剛性マトリックス ek(*,*)
      call gauss2(c,ek,hnum,n,x,detj,nmte,nn)
100  continue
c 要素剛性マトリックスに板厚、もしくは軸対称の係数をかける
      call gauss3(nmte,ek,t)
c element stiffness --> global stiffness 要素→全体剛性への埋め込み
      call merge(ek,nn,gk,nmte)
c clear element stiffness matrix 要素剛性マトリックスの初期化 Fortran90 では単に、ek=0.0d0 と書けばよい。
      call clear2(ek,nmte)
150  continue
c boundary condition 境界条件の設定
      call bound(ifor,fb,f,idis,ub,gk)
c gauss-erase-method ガウスの消去法
      call solve(gk,f,u)
c strain b*a ひずみの計算
      call strain(nn,u,ep,bm,nmte,ngng)
c stress d*strain 応力の計算
      call stress(d,ep,sg,ngng)
c output 出力 result.dat
      call output(u,ep,sg,ngng,neus,nnus)
c output(average at node) 節点平均量の出力 result.dat
      call output_av(u,ep,sg,ngng,nmte,neus,nnus,nn)
c
      close(2)
c
      stop
      end

```

1.1 モデル

平面ひずみ・平面応力・軸対称問題に対応している。

1.2 物性値 (弾性定数)

ヤング率・ポアソン比を与える等方弾性体のみでなく、直交異方性が扱えるようにした (データフォーマット参照)。2次元における弾性定数については、巻末参照

1.3 要素

以下のアイソパラメトリック要素が扱える。

- 四辺形 4 節点要素
- 四辺形 8 節点要素
- 三角形 6 節点要素

ガウス積分は2次と3次が可能になっている。節点番号と積分点番号の並びを図3に示す。これらの並びは汎用コードABAQUSと同じになるように変更した。

1.4 荷重

荷重は、それぞれの自由度に与える方式になっている。2次のメッシュの場合、これは複雑な作業となるため、この作業を行うソフトを開発した。

以下にプログラム名と結果を示す。

- 分布荷重

```
dload.f
--> distributed load (2D version - rectangle)
presst=1.0
pressn=2.0
NODE 1  0.166667  0.333333
NODE 2  0.666667  1.333333
NODE 3  0.166667  0.333333
dload-tri.f
--> distributed load (2D version - triangle )
presst=1.0
pressn=2.0
NODE 1  0.166667  0.333333
NODE 2  0.666667  1.333333
NODE 3  0.166667  0.333333
```

- 重力荷重

```
gload.f
--> gravity load (2D version - rectangle)
NODE 1  0.  -8.33333E-02
NODE 2  0.  0.333333
NODE 3  0.  -8.33333E-02
NODE 4  0.  0.333333
NODE 5  0.  -8.33333E-02
NODE 6  0.  0.333333
NODE 7  0.  -8.33333E-02
NODE 8  0.  0.333333
gload-tri.f
--> gravity load (2D version - tirangle)
NODE 1  0.  2.42144E-08
NODE 2  0.  0.166667
NODE 3  0.  -1.11759E-08
NODE 4  0.  0.166667
NODE 5  0.  -7.45058E-09
NODE 6  0.  0.166667
```

- 熱応力

```
thload.f
--> initial stress load (2D version - rectangle)
NODE 1  0.166667  0.166667
NODE 2  -8.94070E-08  0.666667
NODE 3  -0.166667  0.166667
```

```

NODE 4 -0.666667 9.68575E-08
NODE 5 -0.166667 -0.166667
NODE 6 -6.70552E-08 -0.666667
NODE 7 0.166667 -0.166667
NODE 8 0.666667 5.96046E-08
thload-tri.f
--> initial stress load (2D version - triangle)
NODE 1 0.166667 0.166667
NODE 2 -1.03687E-07 0.666667
NODE 3 -0.166667 0.
NODE 4 -0.666667 -0.666667
NODE 5 0. -0.166667
NODE 6 0.666667 -5.96046E-08

```

1.5 出力

出力は、変位・積分点応力／ひずみと節点応力／ひずみである。積分点応力・ひずみを外挿して節点の応力・ひずみを求める。

各節点応力を $\sigma_1, \sigma_2, \dots$ とし、積分点応力を $\hat{\sigma}_I, \hat{\sigma}_{II}, \dots$ とすると、積分点応力と節点応力の間には、ひとつの要素内で式 (1) の関係が成り立つ。関係式は積分点の数だけ定義する。 s_i は積分点の局所座標である。 N は形状関数である。

$$\begin{aligned}
 \sum_i \sigma_i N_i(s_I) &= \hat{\sigma}_I \\
 \sum_i \sigma_i N_i(s_{II}) &= \hat{\sigma}_{II} \\
 &\dots\dots\dots
 \end{aligned}
 \tag{1}$$

これは、 $N_i(s)$ を成分とするマトリックス N_s を使って、以下のように書ける。節点応力を求めるためには N_s の逆マトリックスを求め、積分点応力にかければ良い。 σ は σ_i をベクトル化したものである。

$$N_s \sigma = \hat{\sigma} \tag{2}$$

$$\hat{\sigma} = N_s^{-1} \sigma \tag{3}$$

すべての要素において節点応力を求め、重なった節点について平均をとった。

1.6 入力データのチェック

入力データをチェックして”datacheck.dat”に出力する機能を追加した。”datacheck.dat”の例を以下に示す。

```

-----datacheck.dat-----
DATA CHECK
PLANE STRESS ANALYSIS

```

```

THICKNESS =      1.00000000000000
NUMBER OF MATERIAL =      1
INTEGRATION ORDER =      2
Material 1
YOUNG=      210000.00000000, POISSON=      0.40000000000000
TOTAL NODES      =      15
TOTAL ELEMENTS =      8
TOTAL FREEDOMS =      30
ELEMENT 1: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 2: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 3: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 4: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 5: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 6: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 7: MATERIAL 1 : 4 NODES RECTANGLE
ELEMENT 8: MATERIAL 1 : 4 NODES RECTANGLE
TOTAL INT.POINTS =      32
COORDINATE OF NODES
PRO.NUM. , USR NUM. ,COORDINATION OF NODES <--- Program number & User number
 1  2  0.1500000E+01  0.1000000E+01
 2  4  0.2000000E+01  0.1500000E+01
 3  30 0.2000000E+01  0.1000000E+01
 4  1  0.1000000E+01  0.1000000E+01
 5  5  0.2000000E+01  0.2000000E+01
 6  6  0.1500000E+01  0.2000000E+01
 7  7  0.1000000E+01  0.2000000E+01
 8  110 0.1500000E+01  0.3000000E+01
 9  8  0.1000000E+01  0.1500000E+01
10  9  0.2000000E+01  0.2500000E+01
11  100 0.2000000E+01  0.3000000E+01
12  120 0.1000000E+01  0.3000000E+01
13  130 0.1000000E+01  0.2500000E+01
14  140 0.1500000E+01  0.1500000E+01
15  150 0.1500000E+01  0.2500000E+01
*ELEMENT NODE IS RECONSTRUCTED
USER 1 2 140 8 <--- User number
PROGRAM 4 1 14 9 <--- Program number
USER 2 30 4 140
PROGRAM 1 3 2 14
USER 8 140 6 7
PROGRAM 9 14 6 7
USER 140 4 5 6
PROGRAM 14 2 5 6
USER 7 6 150 130
PROGRAM 7 6 15 13
USER 6 5 9 150
PROGRAM 6 5 10 15
USER 130 150 110 120
PROGRAM 13 15 8 12
USER 150 9 100 110
PROGRAM 15 10 11 8
*FORCE BOUNDARY
NODE 100-->FY= -30.000000000000 22 <-- position in global element matrix
NODE 110-->FY= -60.000000000000 16
NODE 120-->FY= -30.000000000000 24

```



```
*DISPLACEMENT BOUNDARY
NODE 1-->UX= 0. 7
NODE 1-->UY= 0. 8
NODE 2-->UY= 0. 2
NODE 30-->UY= 0. 6
BAND HALF WIDTH= 28
```

1.7 その他

要素番号、節点番号がとびとびの値であっても、プログラム内では連続の値に修正して計算し、出力は入力と同じ番号で行う機能を追加した。逆行列計算はバンド構造を考慮したガウスの消去法を用いているため、大きな問題には適用できない。

HISAK input data format

S.Izumi 98-06-09

3	—————>	NGAUSS : Order of Gauss integration
0,1,0,2	—————>	NPARA : analysis condition , T : Thickness, MNUM : Number of Materials
1	—————>	IECON(I) : Elastic properties of material
1,2,1000,0,0,4		1 : isotropic , 2 : orthotropic
1		case1) Number , Young modulus , Poisson ration
2,2,100,0,0,4		case2) Number , C11,C22,C33,C44,C55,C66,C12,C13,C23
14,3	—————>	NTN : Total number of nodes , NTE : Total number of elements
1,1,6	—————>	Number , NOM : Material number , NN : Number of nodes in a element
1,2,3,4,5,6	—————>	NN(I) : Node number of a element
2,1,6		
1,6,5,9,8,7		
3,1,8		
8,9,5,10,11,12,13,14		
1,1,0,0,0	—————>	NUMBER , X-coordinate , Y-coordinate
2,1,5,0,0		
3,2,0,0,0		
4,2,0,1,25		
5,2,0,1,5		
6,1,5,1,25		
7,1,0,1,25		
8,1,0,1,5		
9,1,5,1,5		
10,2,0,1,75		
11,2,0,2,0		
12,1,5,2,0		
13,1,0,2,0		
14,1,0,1,75		
3	—————>	NTFOR : Number of freedom which constrain the load value
10,2,0.166666667	—————>	NODE , FREEDOM , LOAD
13,2,0.166666667		IFOR : Freedom number , FB : Load value
22,2,0.666666667		IFOR = (NODE - 1) * IDIM + FREEFOM , FB =LOAD
4	—————>	NTDIS : Number of freedom which constrain the displacement
1,1,0,0	—————>	NODE , FREEDOM , DISPLACEMENT
1,2,0,0		IDIS : Freedom number , UB : initial displacement value
2,2,0,0		IDIS = (NODE - 1) * IDIM + FREEFOM , UB=DISPLACEMENT
3,2,0,0		

図 2: The format of the input data

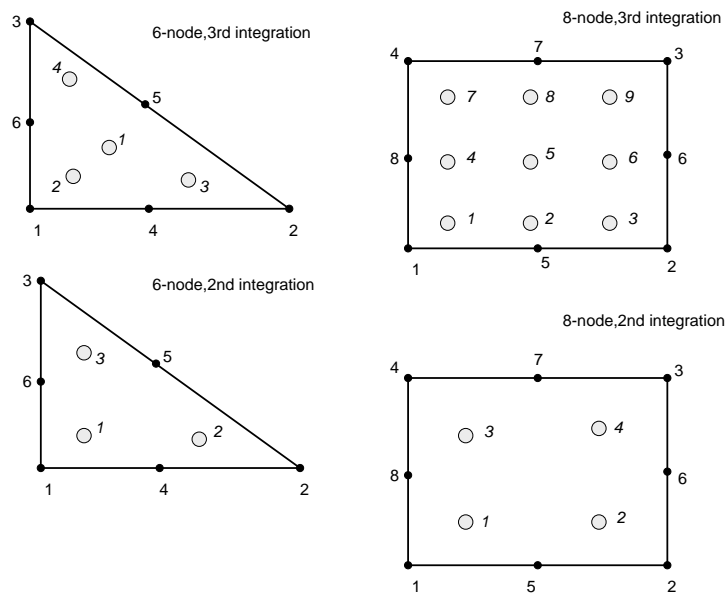


图 3: node number and integration point number on the various elements

2 三次元シミュレータ

三次元解析のフローチャートは二次元と同様であるが、入力データに少しの変更がある。三次元解析の入力データのフォーマットを示す。また、ソルバはガウスの消去法、コレスキー法、共役勾配法に対応し、EQUATOION(ABAQUSでの名称、ある節点の変位を基準節点の変位と同じに拘束する機能) 機能、無限要素が追加されている。

共役勾配法ソルバのプログラムの実行方式は、カレントの下に directory(任意設定、デフォルト files) を作って、そこに input.dat を置く。実行は

```
2      "NGAUSS : ガウス積分次数
2      "MNUM : 材料の種類の数"
2      ECON(I) : 材料の弾性的性質番号 1:等方体, 2:直交異方性"
1,167.4,167.4,167.4,79.6,79.6,79.6,65.2,65.2,65.2 "材料番号, C11,C22,C33,C44,C55,C66,C12,C13,C23"
1
2,21000.0,0.4 "材料番号, ヤング率, ポアソン比"
27 8      "NTN : 全節点数, NTE : 全要素数"
1, 2, 8, 0 "要素番号, 材料番号, 節点数, 無限要素用フラグ (1 : 無限要素)
1, 2, 11, 10, 4, 5, 14, 130 "NN(I) : 要素の節点番号"
20, 2, 8, 0
2, 3, 12, 11, 5, 6, 15, 14
3, 2, 8, 0
4, 5, 14, 130, 7, 8, 17, 16
4, 2, 8, 0
5, 6, 15, 14, 8, 9, 18, 17
5, 2, 8, 0
10, 11, 20, 19, 130, 14, 23, 22
6, 2, 8, 0
11, 12, 21, 20, 14, 15, 24, 23
7, 2, 8, 0
130, 14, 23, 22, 16, 17, 26, 25
8, 2, 8, 0
14, 15, 24, 23, 17, 18, 27, 26
1, 0.000000E+00, 0.000000E+00, 0.000000E+00 "節点番号, XYZ 座標"
2, 0.500000E+00, 0.000000E+00, 0.000000E+00
3, 0.100000E+01, 0.000000E+00, 0.000000E+00
4, 0.000000E+00, 0.000000E+00, 0.500000E+00
5, 0.500000E+00, 0.000000E+00, 0.500000E+00
6, 0.100000E+01, 0.000000E+00, 0.500000E+00
7, 0.000000E+00, 0.000000E+00, 0.100000E+01
8, 0.500000E+00, 0.000000E+00, 0.100000E+01
9, 0.100000E+01, 0.000000E+00, 0.100000E+01
10, 0.000000E+00, 0.500000E+00, 0.000000E+00
11, 0.500000E+00, 0.500000E+00, 0.000000E+00
12, 0.100000E+01, 0.500000E+00, 0.000000E+00
130, 0.000000E+00, 0.500000E+00, 0.500000E+00
14, 0.500000E+00, 0.500000E+00, 0.500000E+00
15, 0.100000E+01, 0.500000E+00, 0.500000E+00
16, 0.000000E+00, 0.500000E+00, 0.100000E+01
17, 0.500000E+00, 0.500000E+00, 0.100000E+01
18, 0.100000E+01, 0.500000E+00, 0.100000E+01
19, 0.000000E+00, 0.100000E+01, 0.000000E+00
20, 0.500000E+00, 0.100000E+01, 0.000000E+00
21, 0.100000E+01, 0.100000E+01, 0.000000E+00
22, 0.000000E+00, 0.100000E+01, 0.500000E+00
```

```

23, 0.5000000E+00, 0.1000000E+01, 0.5000000E+00
24, 0.1000000E+01, 0.1000000E+01, 0.5000000E+00
25, 0.0000000E+00, 0.1000000E+01, 0.1000000E+01
26, 0.5000000E+00, 0.1000000E+01, 0.1000000E+01
27, 0.1000000E+01, 0.1000000E+01, 0.1000000E+01
1      "NTFOR : 荷重を負荷する自由度の個数"
27,3, 1.0  "節点, 自由度 (1~3), 荷重"(IFOR : 自由度, FB : 荷重)
27      "NTDIS : 変位拘束する自由度の個数"
1, 1, 0.0000000E+00 "節点, 自由度, 変位"(IDIS : 自由度, UB : 変位)
4, 1, 0.0000000E+00
7, 1, 0.0000000E+00
10, 1, 0.0000000E+00
130,1, 0.0000000E+00
16, 1, 0.0000000E+00
19, 1, 0.0000000E+00
22, 1, 0.0000000E+00
25, 1, 0.0000000E+00
1, 2, 0.0000000E+00
2, 2, 0.0000000E+00
3, 2, 0.0000000E+00
4, 2, 0.0000000E+00
5, 2, 0.0000000E+00
6, 2, 0.0000000E+00
7, 2, 0.0000000E+00
8, 2, 0.0000000E+00
9, 2, 0.0000000E+00
1, 3, 0.0000000E+00
2, 3, 0.0000000E+00
3, 3, 0.0000000E+00
10, 3, 0.0000000E+00
11, 3, 0.0000000E+00
12, 3, 0.0000000E+00
19, 3, 0.0000000E+00
20, 3, 0.0000000E+00
21, 3, 0.0000000E+00
1      "EQUATION をセットする数"
81,8      "元節点, コピー節点の数"
21,24,27,48,51,54,75,78 "コピー節点"

```

共役勾配法ソルバのみ最後に、計算の設定が必要

```

1.5,0.1,1.0E-6 ! 1.0E-6 は収束の精度
100,0,0,0      ! 100 はイタレーションの回数

```

2.1 要素

以下のアイソパラメトリック要素が扱える。

- 三角柱 6 節点要素 (積分次数 1,2)
- 三角柱 15 節点要素 (積分次数 2,3)
- 六面体 8 節点要素 (積分次数 2,3)

● 六面体 20 節点要素 (積分次数 2,3)

節点番号と積分点番号の並びは汎用コード ABAQUS と同じである。

例として六面体 8 節点要素の形状関数 N_i を式 (4) に示す。2 次の場合の積分点は、規格化された座標系で $\mathbf{s} = (\xi, \eta, \zeta) = (\pm\frac{\sqrt{3}}{3}, \pm\frac{\sqrt{3}}{3}, \pm\frac{\sqrt{3}}{3})$ に位置する。図 4 に節点・積分点の位置を示す。

ただし、八面体 8 節点要素に限って無限要素が使用できるようにしてある。要素番号の指定の行において、4 列目に 1 を追加する (プログラムは共役勾配法ソルバーのもののみに対応している)。

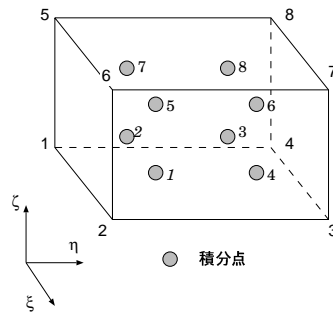


図 4: A hexahedral 8-node isoparametric element

$$\begin{aligned}
 N_1 &= \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta), N_2 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta) \\
 N_3 &= \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta), N_4 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) \\
 N_5 &= \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta), N_6 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) \\
 N_7 &= \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta), N_8 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)
 \end{aligned}$$

1

2.2 Equation 機能の追加

以下の式のように、ある節点の j の変位と等しい変位をとる節点の組 ($i_1 \sim i_n$ など) がある場合のみ対応。

$$\begin{aligned}
 U_{i_1} - U_j &= 0 \\
 U_{i_2} - U_j &= 0 \\
 U_{i_3} - U_j &= 0 \\
 &\dots\dots
 \end{aligned} \tag{4}$$

$$U_{i_n} - U_j = 0 \tag{5}$$

(6)

¹NIF(INMTE),DXIF(INMTE),DETIF(INMTE),DZEIF(INMTE) を追加した。

入力方法は、input ファイルの最後に以下のような文を追加する (入力は自由度)

```
-----
1      <--- equation の数 (5 個まで)
81,8   <--- j の ,i?の個数 (100 個まで)
21,24,27,48,51,54,75,78 <-- i1,i2,i3,...
-----
```

プログラムの修正点は、巻末に記す。

3 連立一次方程式の解法

有限要素法では、最終的に以下の連立一次方程式 $\mathbf{K}\mathbf{a} = \mathbf{f}$ を解くことになる。ここで、 \mathbf{K} :全体剛性マトリックス、 \mathbf{a} :変位ベクトル、 \mathbf{f} :力ベクトルである。

作成したプログラムは以下の 4 種類の解法が可能である。

- ガウスの消去法
- 修正コレスキー法による LU 分解
- 共役勾配法
- スカイライン法

3.1 修正コレスキー法による LU 分解 (三次元シミュレータ)

反復計算においては、力ベクトル \mathbf{f} が頻繁に変わることが予想されるため本手法をが必要である。本手法では剛性マトリックス \mathbf{K} を下三角行列 \mathbf{L} と対角行列 \mathbf{D} を用いて $\mathbf{K} = \mathbf{L}\mathbf{D}\mathbf{L}^t$ の形に分解する。 t は転置を表す。このような分解が可能ならば、連立方程式の解は

$$\mathbf{L}\mathbf{D}\mathbf{z} = \mathbf{f}, \quad \mathbf{L}^t\mathbf{a} = \mathbf{z} \tag{7}$$

を解くことによって得られる²。 D_{ij}, L_{ij} は $i \geq j$ で $L_{ij} \neq 0, L_{ii} = 1$ となることに注意して、

$$K_{ii} = \sum_{j=1}^i L_{ij}D_{jj}L_{ij} = L_{ii}^2D_{ii} + \sum_{j=1}^{i-1} L_{ij}^2D_{jj} \tag{8}$$

$$K_{ik} = \sum_{k=1}^j L_{jk}D_{kk}L_{jk} = L_{ij}D_{jj}L_{jj} + \sum_{k=1}^{j-1} L_{ik}D_{kk}L_{jk} \tag{9}$$

の関係式より、それぞれ以下のように求めることができる。

²前者は前進代入、後者は後退代入によって容易に計算可能である

$$D_{ii} = K_{ii} - \sum_{j=1}^{i-1} L_{ij}^2 D_{jj} \quad (10)$$

$$L_{ij} = \left(K_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_{kk} \right) / D_{jj} \quad (11)$$

コレスキー法のプログラムが入ったディレクトリの solve_c.f で、剛性マトリクスがプログラムの形式にコピーされ、修正コレスキー分解 (lisb.f) が行われる。solve_cd.f では、前進代入法 (sbsub.f) により変位が求められる。これは無駄が多いので修正されるべき点である。

3.2 共役勾配法による解法（三次元シミュレータ）

修正コレスキー法などの直接法に対し、共役勾配法を使った手法を反復法と呼ぶ。反復法の利点は、剛性マトリクスの非ゼロ成分のみをデータとして持てばよいのでメモリーが少ない、問題によってはかなり高速に解けるという点である。

以下の変数を追加した。

```
ccc FOR ICCG METHOD
    DIMENSION NSMALL(INODE), NSNODE(INODE, INMTE*10)
    DIMENSION INDEX2(IGK), INDEX5(INODE*INMTE*10), NSUM(INODE)
    DIMENSION SYK(IBN*IGK)
    DIMENSION WK1(IGK), WK2(IGK), WK3(IGK), WK4(IGK), WK5(IGK), WK6(IGK)
    DIMENSION IWK(IGK)
c NSMALL(I) : see data.f
```

solve_c.f でデータを 1 次元に加工し、必要な配列情報を作成する。solve_cg.f で femcg3.f を呼び出して計算している。詳細は参考文献 [3] の p317-320 にのっている。

3.3 スカイライン法

プログラムのみ作成してある。専用マニュアル参照。

4 † 弾性定数とヤング率

弾性定数を 2 次元系で入力する方法について述べる。

4.1 Orthotropic(Orthorhombic)

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{pmatrix} \quad (12)$$

- 平面応力

$\sigma_z = \tau_{xz} = \tau_{yz} = 0$ なので、

$$C_{13}\varepsilon_x + C_{23}\varepsilon_y + C_{33}\varepsilon_z = 0 \quad (13)$$

$$\varepsilon_z = -\frac{1}{C_{33}}(C_{13}\varepsilon_x + C_{23}\varepsilon_y) \quad (14)$$

これらを式 (12) に代入する。 ε_z は式 (14) を使う。

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} & 0 \\ D_{12} & D_{22} & 0 \\ 0 & 0 & D_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (15)$$

$$D_{11} = \frac{1}{C_{33}}(C_{11}C_{33} - C_{13}^2) \quad (16)$$

$$D_{12} = \frac{1}{C_{33}}(C_{12}C_{33} - C_{13}C_{23}) \quad (17)$$

$$D_{22} = \frac{1}{C_{33}}(C_{22}C_{33} - C_{23}^2) \quad (18)$$

$$D_{44} = C_{44} \quad (19)$$

- 平面ひずみ

$\varepsilon_z = \gamma_{xz} = \gamma_{yz} = 0$ なので、

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (20)$$

$$\sigma_z = C_{13}\varepsilon_x + C_{23}\varepsilon_y \quad (21)$$

4.2 Cubic

$$C_{11} = C_{22} = C_{33}, C_{12} = C_{13} = C_{23}$$

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{pmatrix} \quad (22)$$

- 平面応力

$\sigma_z = \tau_{xz} = \tau_{yz} = 0$ なので、

$$C_{12}\varepsilon_x + C_{12}\varepsilon_y + C_{11}\varepsilon_z = 0 \quad (23)$$

$$\varepsilon_z = -\frac{C_{12}}{C_{11}}(\varepsilon_x + \varepsilon_y) \quad (24)$$

これらを式 (22) に代入する。 ε_z は式 (24) を使う。

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} & 0 \\ D_{12} & D_{11} & 0 \\ 0 & 0 & D_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (25)$$

$$D_{11} = \frac{1}{C_{11}}(C_{11}^2 - C_{12}^2) \quad (26)$$

$$D_{12} = \frac{1}{C_{11}}(C_{12}C_{11} - C_{12}^2) \quad (27)$$

$$D_{44} = C_{44} \quad (28)$$

- 平面ひずみ

$\varepsilon_z = \gamma_{xz} = \gamma_{yz} = 0$ なので、

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{11} & 0 \\ 0 & 0 & C_{44} \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (29)$$

$$\sigma_z = C_{12}(\varepsilon_x + \varepsilon_y) \quad (30)$$

- ヤング率との関係

$$S_{11} = \frac{C_{11} + C_{12}}{(C_{11} - C_{12})(C_{11} + 2C_{12})}, \quad S_{12} = \frac{-C_{12}}{(C_{11} - C_{12})(C_{11} + 2C_{12})}, \quad S_{44} = \frac{1}{S_{44}} \quad (31)$$

$$C_{11} = \frac{S_{11} + S_{12}}{(S_{11} - S_{12})(S_{11} + 2S_{12})}, \quad C_{12} = \frac{-S_{12}}{(S_{11} - S_{12})(S_{11} + 2S_{12})}, \quad C_{44} = \frac{1}{S_{44}} \quad (32)$$

$$E = \frac{1}{S_{11}}, \quad \nu = \frac{-S_{12}}{S_{11}}, \quad G = \frac{1}{S_{44}} \quad (33)$$

これらを代入すると

- 平面応力

$$D_{11} = \frac{E}{1 - \nu^2}, \quad D_{12} = \frac{E\nu}{1 - \nu^2}, \quad D_{44} = G \quad (34)$$

$$\varepsilon_z = \nu(\sigma_x + \sigma_y) \quad (35)$$

- 平面ひずみ

$$C_{11} = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)}, \quad C_{12} = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}, \quad C_{44} = G \quad (36)$$

$$\sigma_z = -\frac{\nu}{E}(\sigma_x + \sigma_y) \quad (37)$$

4.3 Isotropic

$$C_{44} = \frac{C_{11} - C_{12}}{2}, \quad G = \frac{E}{2(1 + \nu)} \quad (38)$$

- 平面応力

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \frac{E}{1 - \nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1 - \nu)/2 \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (39)$$

$$\varepsilon_z = \nu(\sigma_x + \sigma_y), \quad \sigma_z = 0 \quad (40)$$

- 平面ひずみ

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1 & \nu(1-\nu) & 0 \\ \nu(1-\nu) & 1 & 0 \\ 0 & 0 & (1-2\nu)/2(1-\nu) \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (41)$$

$$\sigma_z = -\frac{\nu}{E}(\sigma_x + \sigma_y) , \quad \varepsilon_z = 0 \quad (42)$$

5 †Equation 機能の追加に伴うプログラム修正点（三次元シミュレータ）

- const.inc

```
cccc FOR EQUATION
c  NAEQ : NUMBER OF EQUATION
c    NEQB : ROOT FREEDOM
c    NTEQ : NUMBER OF COPY FREEDOM
c    NEQ  : COPY FREEDOM
      COMMON /EQ0/ NAEQ
      COMMON /EQ1/ NEQB(5),NTEQ(5)
      COMMON /EQ2/ NEQ(5,100)
```

- data.f に入力部追加

```
      READ(1,*)NAEQ
      IF (NAEQ.EQ.0) goto 88
      DO 80 I = 1,NAEQ
      READ(1,*)NEQB(I),NTEQ(I)
      READ(1,*)(NEQ(I,J),J=1,NTEQ(I))
80    CONTINUE
88    CONTINUE
```

- band.f

equation を使った場合にバンド幅が広がることに対応した修正

- merge.f

全体剛性マトリクスの埋め込み箇所の修正

- bound.f

カベクトルの修正

- output.f

equation で結ばれた節点の変位を等しくする修正

6 † 要素内局所座標の算出

FEM-MD 結合手法において、原子を FEM メッシュに埋め込む手法を使う。要素内のある点の位置を、その要素の局所座標系 (ξ, η, ζ) で表す手法を示す。要素の節点の座標値を (x_i, y_i, z_i) (i は 1~8) とすると、注目する点の座標 (x_a, y_a, z_a) は、その点の局所座標値 (ξ_a, η_a, ζ_a) を使えば、以下の式で表すことができる。

$$\begin{aligned} x_a &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) x_i \\ y_a &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) y_i \\ z_a &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) z_i \end{aligned} \quad (43)$$

式 (43) を以下のように書き直す。

$$\begin{aligned} f_x(\xi_a, \eta_a, \zeta_a) &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) x_i - x_a = 0 \\ f_y(\xi_a, \eta_a, \zeta_a) &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) y_i - y_a = 0 \\ f_z(\xi_a, \eta_a, \zeta_a) &= \sum_{i=1}^8 N_i(\xi_a, \eta_a, \zeta_a) z_i - z_a = 0 \end{aligned} \quad (44)$$

この連立一次方程式を解くために式 (46) の $(\Delta\xi, \Delta\eta, \Delta\zeta)$ を使った反復計算 (式 (47)) を行った。

$$A = \begin{pmatrix} \frac{\partial f_x}{\partial \xi} & \frac{\partial f_x}{\partial \eta} & \frac{\partial f_x}{\partial \zeta} \\ \frac{\partial f_y}{\partial \xi} & \frac{\partial f_y}{\partial \eta} & \frac{\partial f_y}{\partial \zeta} \\ \frac{\partial f_z}{\partial \xi} & \frac{\partial f_z}{\partial \eta} & \frac{\partial f_z}{\partial \zeta} \end{pmatrix} \quad (45)$$

$$\begin{Bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{Bmatrix} = -A^{-1} \begin{Bmatrix} f_x(\xi_k, \eta_k, \zeta_k) \\ f_y(\xi_k, \eta_k, \zeta_k) \\ f_z(\xi_k, \eta_k, \zeta_k) \end{Bmatrix} \quad (46)$$

$$\begin{aligned} \xi_{k+1} &= \xi_k + \Delta\xi \\ \eta_{k+1} &= \eta_k + \Delta\eta \\ \zeta_{k+1} &= \zeta_k + \Delta\zeta \end{aligned} \quad (47)$$

参考文献

- [1] Zienkiewicz, O.C., Taylor, R.L., マトリックス有限要素法, (1996), 科学技術出版者
- [2] G. ガッド, G. トゥゾー, 有限要素法全解, パーソナルメディア
- [3] 小国 力, 行列計算ソフトウェア, (1991), 173, 丸善.