I －28

マイコンによる骨組構造の境界要素法（その１）

第一工業大学　正員　塚本 正文

1 まえがき

載荷された骨組構造の支点および求めようとする変位の所定格点を境界点とし，他の諸格点を内点とする。有限要素法の手法により内点および境界点の剛性方程式をそれぞれ作製する。　内点剛性方程式から内点変位を境界点変位と内点外力で表わし，これを境界点剛性方程式に代入し，境界点外力を境界点変位と内点外力の式に変換する。　この式から境界点の支点反力および境界点変位を算定する。　すなわち有限要素法の手法を拡張して所定の境界点変位と支点反力を算定することができる。

この算法を利用するに当って，第1の内点変位から最終内点変位の式を逐次境界点剛性方程式に前進消去させることによりマイコンの泣きどころであるメモリーを大いに軽減できる。

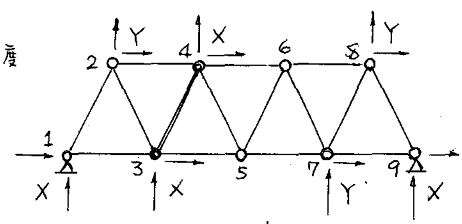
2 算法の原理

図－1のトラスでは支点1，9およびもとめようとする変位の格点3，4が境界点で他の格点は内点である。3，4の格点変位が算定されると部材34の部材力がきまる。

NB　（境界点の数）× 自由度　　NI　（内点の数）× 自由度

X　境界点の外力　　Y　内点の外力

U　境界点の変位　　V　内点の変位



図－1

内点の剛性方程式　　$\mathbb{Y} = \mathbb{B}\,\mathbb{V} + \mathbb{C}\,\mathbb{U}$　---------- (1)

境界点の剛性方程式　　$\mathbb{X} = \mathbb{E}\,\mathbb{U} + \mathbb{F}\,\mathbb{V}$　---------- (2)

$\mathbb{B}$はNI元，$\mathbb{C}$はNI×NB元，$\mathbb{E}$はNB元，$\mathbb{F}$はNB×NI元のマトリックス，$\mathbb{Y}$，$\mathbb{V}$はNI次，$\mathbb{X}$，$\mathbb{U}$はNB次のヴェクトルである。　これらの方程式は骨組構造の図を見ながら自動的にそれぞれプログラミングできる。

(1)式より　　$\mathbb{V} = -\mathbb{B}^{-1}\mathbb{C}\,\mathbb{U} + \mathbb{B}^{-1}\mathbb{Y} = \mathbb{D}\,\mathbb{U} + \mathbb{B}^{-1}\mathbb{Y}$　------------ (3)

(3)式を(2)式に代入すると

$$\mathbb{X} = \mathbb{E}\,\mathbb{U} + \mathbb{F}\cdot\mathbb{D}\,\mathbb{U} + \mathbb{F}\cdot\mathbb{B}^{-1}\mathbb{Y}$$

$\mathbb{G} = \mathbb{E} + \mathbb{F}\cdot\mathbb{D}$　　$\mathbb{H} = \mathbb{F}\cdot\mathbb{B}^{-1}$　　とすると　------- (4)

$\mathbb{X} = \mathbb{G}\,\mathbb{U} + \mathbb{H}\,\mathbb{Y}$　------------ (5)

(5)式より，拘束支点の反力X（変位U＝0），所定内点の変位Uを拘束支点以外の外力X，内点の外力Yを用いて算定することができる。　計算途中の$\mathbb{B}$，$\mathbb{E}$，$\mathbb{D}$，$\mathbb{G}$，$\mathbb{H}$の各要素は構造原理による所要の条件を満足することはもちろんである。　マイコンのメモリーがゆるされればこの算法どおりの演算によって所要の解を得る。

3 マイコンによる一般的算法

マイコンのメモリーを節約するため以下のように(1)式から $v_1, v_2 \cdots v_{NI}$ を(4)式に逐次前進消去することによって(5)式を作製することができる。(5)式の i 格点の境界点外力 X_i は

$$X_i = [g_{i\cdot1}\ \ g_{i\cdot2}\ \cdots\ g_{i\cdot NB}]\begin{Bmatrix}u_1\\ \vdots\\ u_{NB}\end{Bmatrix} + [h_{i\cdot1}\ \ h_{i\cdot2}\cdots h_{i\cdot NI}]\begin{Bmatrix}v_1\\ v_2\\ \vdots\\ v_{NI}\end{Bmatrix} \quad \text{----------} \quad (1\cdot1)$$

g, h は $\mathbb{E}$, $\mathbb{F}$ の要素である

STEP 1

(1)式の内点1の剛性方程式より右辺の v_1 を算定する

$$v_1 = [d_{1\cdot1}\ \ d_{1\cdot2}\ \cdots\ d_{1\cdot NI}]\begin{Bmatrix}Y_1\\ v_2\\ \vdots\\ v_{NI}\end{Bmatrix} + [b_{1\cdot1}\ \ b_{1\cdot2}\ \cdots\ b_{1\cdot NB}]\begin{Bmatrix}u_1\\ u_2\\ \vdots\\ u_{NB}\end{Bmatrix} \quad \text{----------} \quad (2\cdot1)$$

FILE [v_1]

(2・1) を (1・1) に代入して

$$X_i = [g_{i\cdot1}\ \ g_{i\cdot2}\ \cdots\ g_{i\cdot NB}]\begin{Bmatrix}u_1\\ \vdots\\ u_{NB}\end{Bmatrix} + [h_{i\cdot1}\ \ h_{i\cdot2}\cdots h_{i\cdot NI}]\begin{Bmatrix}Y_1\\ v_2\\ \vdots\\ v_{NI}\end{Bmatrix} \quad \text{----------} \quad (1\cdot2)$$

ここに $g_i \leftarrow g_i + h_{i\cdot1} b_{1\cdot i}$　　$h_i \leftarrow h_i + h_{i\cdot1} d_{1\cdot i}$ $(i \neq 1)$　　$h_1 \leftarrow h_{1\cdot1} d_{1\cdot1}$

STEP 2

前段同様，内点2の剛性方程式より v_2 を算定する。次に CALL [v_1] し，

$$v_1 = [d_{2\cdot1}\ \ d_{2\cdot2}\ \cdots\ d_{2\cdot NI}]\begin{Bmatrix}Y_1\\ v_2\\ \vdots\\ v_{NI}\end{Bmatrix} + [b_{2\cdot1}\ \ b_{2\cdot2}\ \cdots\ b_{2\cdot NB}]\begin{Bmatrix}u_1\\ u_2\\ \vdots\\ u_{NB}\end{Bmatrix}$$

d, b は添字2の係数として呼び出す（INPUT）。これを v_2 の式に代入すると

$$v_2 = [d_{1\cdot1}\ \ d_{1\cdot2}\ \cdots\ d_{1\cdot NI}]\begin{Bmatrix}Y_1\\ Y_2\\ \vdots\\ v_{NI}\end{Bmatrix} + [b_{1\cdot1}\ \ b_{1\cdot2}\ \cdots\ b_{1\cdot NB}]\begin{Bmatrix}u_1\\ u_2\\ \vdots\\ u_{NB}\end{Bmatrix} \quad \text{----------} \quad (2\cdot2)$$

(2・2) を (1・2) に代入する。

FILE [v_2]

以下同様の操作をくり返すと(5)式が算定される。各内点の変位 v を知りたいときは FILE された v を後退代入することによって求めることができる。

4 むすび

上記 3 の算法によれば $\mathbb{G}$ は NB×NB，$\mathbb{H}$ は NB×NI，$\mathbb{D}$ は 2×NI，$\mathbb{B}$ は 2×NB 元のマトリックスですみ，一般的な剛性マトリックスの元数（NB＋NI）×（NB＋NI）元にくらべいちじるしいメモリーの節約となる。この算法の問題点は（2・1），（2・2）… の前進消去とこれを（1・1），（1・2）… への代入演算に時間を要することであろう。次の報告では実際の演算時間について述べたいと思う。